CS7641 ML Lectures - Reinforcement Learning

I. Markov Decision Processes

A. Introduction

Hi Michael. Hey Charles. How's it going? It is going quite well. Thank you very much for asking. How are things going with you? I can't complain because I've been barred by a, a judge. [LAUGH] I'm pretty sure you can still complain. Complaining is a fine art that requires lots of practice. Guess what we're going to do today? I'm reading decision making and reinforcement learning which is very exciting. It is. Except the hyphen. Except for the hyphen? Yeah the hyphens wrong. Why? because it's not decision dash making. You're right. I normally don't do that but people always want to put a dash so what I'm going to do is for your Michael. Just for you. I'm going to remove the dash. Thank you. And I'm going to put it over here where it belongs. [LAUGH] No. There. Well, Michael, this is the first of our discussions on the last mini course of machine learning, Reinforcement Learning, which I believe you know a little bit about. I am very interested in Reinforcement Learning. Excellent. So I'm just going to do a little bit of background. It's going to be relatively short and straight forward. We're going to do a couple of quizzes, because I know how much you like quizzes. And then we're just going to go back and forth and see what we get to learn about Reinforcement Learning. Sound good? Excellent. Excellent, okay so, let's get started.

B. The World 1 Question

Michael, here is a world. In fact, for the purposes of this discussion it is the world. Okay. So imagine the entire universe is well described by this picture over here. Okay? Now this is called a grid world, which is something that people in reinforcement learning love to think about, because it's a nice approximation for all the complexity of the entire universe. Now this particularly world is a three by four grid. So you have one comma one, two comma one, three comma one, four comman one. You have one comma two, one comma three and so on and so forth. For the purpose of this discusision we can think of the world as being a kind of game where you start out in state. Which we're going to call the start state. And your able to execute actions, one of these four, up, down, left to right and the purpose is to wonder around this world in such a way that eventually you make it to the goal over here, this little green spot. You see that Michael? Yep. And under all circumstances, you must avoid the red spot. No. Exactly. Now for this particular example, up does what you think it does, down does what you think it does, as do left and right. But if you find yourself at a boundary, such as right, up here in thi upper-left-hand corner, and you try to go up, you just stay where you are. If you try to go left, you just stay where you are. But if you go right, you do actually end up in the next square. Got it? Think so. Okay. Three last things. One, this little black, space here, is a place you can't enter into, so it acts just like a wall. This green space is the goal, and once you're there it's over. The world is over and you get to start over again. Hm. And once you enter into the red spot, the world is also over and you have to start over again. So you can't go through the red square to get to the green square. Okay, you got it? Yeah. Excellent. So, here's the quiz. Given this particular world, with the physics I just described to you, and given these actions that you can take, up, down, left and right, what is the shortest sequence of actions that would get us from the start state to the goal state? And you can just type in the words up, down, left and right, separated by spaces, or commas or anything like that. Okay. [LAUGH] Say semicolons are allowed, colons are not. Yeah, good. I am glad you made that distinction. Well, it's an important one to make. Okay, you got it? Yeah, I think so. Alright, cool. So let's see what you get then. Go.

C. The World 1 Solution

Okay Michael, you go an answer for me? This isn't meant to be a hard quiz. Oh good. Cause I actually have two answers for you. Oh, I like that. So I would say right, right, up, up, right. Okay, so, right, right, up, up, right. But I feel like I could have also said up, up, right, right, right. And you could have. Both of those are acceptable. right, right does what you think it does. You move right and then right and then up and then up and then right. And that takes five steps. Or you could have gone up, up, right, right, right, which would also have taken you five steps. So that's pretty easy, right? Yeah, I thought so. Rag. So yes, either of these would be correct. We're going to stick with this one in particular, since they're equal, because we gotta pick one of them. So this does point out something that often in a decision problem like this, there are in fact multiple answers or multiple optima, if I can tie it back to our randomized optimization discussions. Okay, so you got the quiz, you got the world, you understand what you can do here. Yes? Yep. Okay, cool. Now I'm going to throw in a little wrinkle.

D. The World 2 Question

I'm going to change the world just a little bit Michael. Okay, that last one was really easy, and it was easy for a bunch of reasons, but one of the reasons it was easy is that every time it took an action it did exactly what you expected it to do. Now I want to introduce a little bit of uncertainty, or stochasticity into the world. So, let me tell you exactly what that means. When you execute an action, it executes correctly, with probability of 0.8. So 80% of the time, if you go up, it goes up. Assuming you can go up. If it goes, if you say down, it goes down, assuming you can. Left, it goes left; right, it goes right. Got it? Yeah! Now, 20% of the time, the action you take actually causes you to move at a right angle. Now, of course, there are two different right angles you could go to, if you go up, you could go either left or right at a right angle. And so that 20% gets distributed uniformally. Okay? Does that make sense? Yeah, I think so. And so if you, if you're in the start state and you try to go up, then there a 10% chance that you tend to bump into the wall. Yes. And then you just stay where you are I guess.

Right. So if you, if you decide that you'd have x here, you have a 80% chance of moving up, you have a 10% chance of moving to the right. And you have a 10% chance of moving to the left but, of course, you'd bump a wall and you would end up right back where you started. Got it? Yeah. Okay, good. So, here is my quiz question for you Michael. You recall, we came up with two sequences and the one that I decided to keep was up, up, right, right, right. My question to you, is, what is the reliability of the sequence, up up, right right, actually getting you from the start to the goal, given these probabilities, this uncertainty? And so, we're just going to try that sequence, and ask whether or not it actually got us to the goal. Exactly. And when I say, reliability, I really mean, what's the probability of it actually succeeding? Interesting. Okay. Alright. So let's see if we can figure out the answer. You're ready? I'm ready. Alright. Go!

E. The World 2 Solution

Okay Michael do you have an answer? I have an answer indeed. Okay, what's the answer? Okay, maybe I don't have an answer indeed. But I have I have the ability to compute one. Okay. I'm, I'm doing some math. So why don't you walk me through the math you're doing. I got .32776. That is correct Michael. Woo-hoo. That was trickier than I thought. Okay well, show me what you did. Alright, so the first thing I did is I said, okay, well this is not so hard, because, from the start-state, if I execute up, up, right, right, right. Each of those things works the way it's supposed to do independently with probability .8. Yep. And so, .8 raised to the 5th power, gives me the probability that that entire sequence will work as intended. Exactly. And what is .8 to the 5th? Do you know? 32,768. with a decimal point in front of it, because you know, powers of 2. Wow. That is correct. But I notice that 32768 is not 32776. No they differ by a little smidge. Mm-hm. So this is what occured to me next is, is there anyway that you could have ended up falling into the goal from that sequence of, of commands not following the intended path. So since actions can have unintended consequences, as they often do. Mm-hm. I was going to ask okay, so if I go up in the first step, there's a probably .1 that I'll actually go to the right. Yep. From there, if I go up, there's a 10% probability that I'll actually go to the right. Mm-hm. From there, the next thing I do is take the right action, but that can actually go up with some probability, .1. Mm-hm, yep. And then another .1 to get to for the next right action to actually cause an up to happen. Mm-hm. And then finally, that last right might actually execute correctly and bring me into the goal. So I could go underneath the barrier, instead of around the barrier with that same sequence. It just isn't very likely. Okay. Well how unlikely is it? Alright. So I did .1 times .1 times .1 times .1 times .8. huh, so the .1 to the 4 times .8 and that's right so this is the probability that the, 4 of the sequences, 4 of these go wrong. In fact exactly the first 4 go wrong. And the last one goes right. Right. And that's equal to, some very, very small number. And when you add it up. You end up with .32776. In fact that's equal to 0.00008. And that's how you get that number and that's correct. And you'll notice that this this sequence happens to work out to be the second sequence that, we had options for. Yeah, the other thing that took us to the goal, right. Yeah. Was exactly the probability of executing that action executing that sequence of transitions given the first command. Yeah, and I think it would work the other way, too. No, it wouldn't quite work the other way. If you had done the sequence right, right, what is it? Right, right. Up, up right? Yeah. In order for that one to work out, you'd add .8 to the 5 and working. And, for it to work out wrong but work out right, the right would have to send you up and then the ups would have to send you right, and then, yeah, so it would actually work out to be the same. Yeah. So no matter which of the two sequences you came up with, they would have the same probability of succeeding. Neat. Nice. That's actually kind of cool. So, good job, Michael. Very good job on the quiz. A lot of people forget this part. And, in fact, if you forgot that part, and got, just this part right, we actually let you pass. But it was wrong. I was kind of expecting you to get it wrong, but I'm glad you got it right too. Thank you Michael, I appreciate your faith in me. [LAUGH] I wrote the quiz. Actually, I stole this from a book, Oh. which, whose little words are now showing up in front of you, exactly where you can go through the details of this quiz. Okay, alright, Michael. So you might ask yourself why I brought this up, and the reason I brought this up is because, what we did in the first case where we just came up with the sequence up up right right right, is we sort of planned out what we would do in a world where nothing could go wrong. But once we introduced this notion of uncertainty, this, this randomness, this stochasticity, we have to do something other than work out in advance what the right answer is, and then just go. We have to do something a little bit more complicated. Either we have to try to execute these, and then every once in a while, see if we've drifted away. And then re-x, re-plan, come up with a new sequence wherever it is we happen to end up, or we have to come up with some way to incorporate all of these uncertainties and probabilities. So that, we never really have to think, rethink what to do in case something goes wrong. So, what I'm going to do next is I'm going to introduce the framework, that's very common, that people use as a way of capturing this stuff, capturing these uncertainties directly. Okay? Hm. You ready? Yeah. Excellent.

F. Markov Decision Processes 1

So this is the framework that we're going to be using through most of the discussions that we'll be having at least on reinforcement learning. The single agent reinforcement learning, and it's called the Markov Decision Process. This should sound familiar to you, Michael. Well, you did say we're going to talk about decisions. That's true, and we need a process for making decisions. And we're going to introduce something called the Markovian property as a part of this discussion and I'll tell you exactly what that means in a moment. So, I'm just going to write out this frame work and just, and tell you what it is and what the problem it produces for us. And then we're going to start talking about solutions through the rest of the discussion. So a Markov Decision Process tries to capture worlds like this one by dividing up in the following way. We say that there are states. And states are a set of tokens that somehow represent every state, for lack of a better word, that one could be in. So, does that make sense to you, Michael? Yeah, I think so. So what would the states be in the world that we've been playing around in so far? So, the only thing that differs from moment to moment is where, I guess, I am. Mm-hm. Like, which grid, grid position I'm in. Right. So, I feel like each different grid position I could be in is a state, maybe there's a state

for being successfully done or unsuccessfully done? It's possible. But let's stick with the simple one. I like that one because that's really, I think, easy to grasp. So, there are at least, of all the states one could reach, there's, well let's see there's four times three minus one, since you can never reach this state. Although we could say it is a state we just happen to never reach it. So, at most if we just think of this grid literally as a grid there are something like twelve different states. And we can represent these states as their X,Y coordinates, say. We could call this, the start state as say 1,1, which is sort of how I described it earlier. We could describe the goal state as 4,4. And say this is how we describe our states. Or frankly, it doesn't matter. We could call these states 1,2,3 up to 12. Or we could name them Fred and Marcus. It doesn't really matter. The point is that they're states, they represent something, and we have some way of knowing which state we happen to be in. Okay? Sure. Okay.

G. Markov Decision Processes 2

Alright so we've got states. So next is whats called the model or the transition model. Sometimes people refer to it as the transition function, and basically it is a function of three variables. It's a state, an action, which I haven't defined for you yet and another state. In fact since I haven't defined what an action is for you yet, let's skip that and actually do define actions for you. So the third part of our model are actions. Actions are the things that you can do in a particular state. So what would the actions be in this world? Well the four different decisions I could make were the up, down, left and right. Though it's maybe a little confusing because I think those were also the four possible outcomes. That's true. Well no, there are other outcomes. You could stay where you were. Right. Right. So these are actually the actions. These are the things that when I'm in a given state I'm allowed to execute. I can either go up, go down, go left or go right. You'll notice that as I described before there was no option not to move. Mm. But there could have been, and there could have been other actions, like teleport, or there's anything that you can imagine. But the point is that your action set represents all of the things that the agent, the robot, the person, or whatever it is you're trying to model, is allowed to do. Now, in its full, generalized form, we can think of the set of actions that one can take as being a function of state Because there's some stage you can do some things and some stage you can't do those things. But most of the time people just treat it as a set of actions. And the actions that aren't allowable in them particular states have no effect. Alright, so we understand states. They're the things that describe the world. We understand actions. Those are the things that you can do, the commands you can execute when your in particular states. And what a model describes, what the transition model describes is in some sense the rules of the game that you're playing. It's the physics of the world. So it's a function of three variables: a state, an action and another state, which, by the way, could actually be the same state. And what this function produces is the probability That you will end up transitioning the state s prime, given that you were in state s, and you took action a. Got it? I think so. so the s prime is where you end up, and the s, a is what you're given. So it's, so you plug these three in. Oh I see, and you get a probability Mm-hm. But the probabilities have to add up to one if you if you sum it up over all the s primes. Right. That's exactly right. So for example if you think about the deterministic case where there was no noise then this is a very simple model. If I'm in the state the start state. And I take the action up, then what's the probability, I end up in the state immediately above it? Was that a 0.08? No, in the, in, when we first started in the deterministic world. Oh, that was probability one. Right, and what would be the probability, you ended up in the state to the right? Probability zero. Right. In fact, the probability that you end up at any of the other states is zero in the deterministic world. Now what about the case when we were in the non-deterministic world where an action would actually execute faithfully only 80% of the time. If I'm in the start state and I go up, what's the probability that I end up. In the state above. That was 0.8 Was the probability, I end up in the state to the right. That was 0.1. And, what was the probability I end up where I started. That was also 0.1. Right, and 0 everywhere else. And, that's just sort of the way it works. So, the model really is an important thing. And the reason, it's important. Is it really does describe the rules of the game. It tells you what you, what will happen if you do something in a particular place. It captures Everything that you know about the transition u, of the world is what you know about the rules, got it? You called it physics before? I called it the physics of the world. Huh. These are the rules that don't change. But they're very different from real world physics. Well, yeah, although they don't have to be. I mean in some sense, you could argue that a mark off decision process, what we described so far, these three things In fact, do describe the universe, the states are, you know, in the positions of all the atoms. The positions and velocities of all the atoms in the universe. The transition miles as you do, take certain positions in the world whatever they are how the state of the universe changes in response to that. And the actions or whatever those set of actions could be. And it can be probabilistic or it's not probabilistic. It's definitely probabilistic. The transition models are by their very definition probabilistic. Gotcha.

H. Markov Decision Processes 3

Now, I actually snuck something important in here, I actually snuck two things that are important in here. The first is, the, what's called the Markovian property. Do you remember what the Markovian property is Michael? From what? From, I dunno. Actually, where does the Markovian property come from? I'm going to say Russia. Okay, yeah, from the, from the Russian. Yeah, so I have Russian ancestors, and they passed onto me this idea that. Markov means that you don't have to condition on anything past the most recent state. That's exactly right. The Markovian property is that only the present matters. And they had to pass that down to me you know, one generation at a time, because you know, Markov property. Exactly right, that was very good Michael. So what does this mean? What this means is our transition function, which shows you the probability you end up in some state as prime, given that you're in state S and took action A, only depends upon the current state S. If it also depended upon where you were 20 minutes before then, you would have to have more S's here. And then you would be violating the Markovian property. So is this like, do historians hate this? Well, you know, one never learns anything from history. No, you're supposed to learn from history, or you're doomed to, I don't know, let me make

something up, repeat it. [LAUGH] Fair enough. Historians probably don't like this, but there is a way for mathematicians to convince them that they're okay with it. And the way that, you, mathematicians convince you that you're okay with this is to point out that you can turn almost anything, into a Markovian Process by simply making certain that your current state remembers everything you need to remember from the past. I see. So, in general, even if something isn't really Markovian, you need to know what you were, not only what you're doing now, but what you were doing five minutes ago. You could just turn your current state into what you're doing now and what you were doing five minutes ago. The obvious problem with that, of course, is that if you have to remember everything from the beginning of time. You're only going to see every state once and it's going to be very difficult to learn anything. But, that Markovian property turns out to be, turns out to be important and it actually allows us to solve these problems in a tractable way. But I snuck in something else, Michael. What I snuck in is this idea about the transition model, is that nothing ever changes. So this second property that matters for Markov decision processes, at least for the sets of things that we're going to be talking about in the beginning, is that things are stationary. That is these for the purposes of this discussion, it means that these rules don't change. Over time. That's one notion of stationary, okay? Does that mean that the agent can never leave the start state? No, the agent can leave the start state any time it takes the action, then it gets another start state. Then how is it, how is it stationary, then? It's not stationary, the world is stationary. The, the transition model is stationary, the physics are stationary, the rules don't change any. The rules don't change. Right. I, I, okay. I see. Then there's another notion of stationary that we'll see a little bit later. Okay, last thing to point out about the definition of a mark on decision process, is the notion of reward. So, reward is simply a scalr value, that you get for being in a state. So for example we might say, you know, this green goal is a really good goal. And so, if you get there, we're going to give you a dollar. This red dual, on the other hand, is a very, very bad state. We don't want you to end there. And so, if you end up there we're going to take away a dollar from you. What if I don't have a dollar? Someone will give you the dollar. The universe will give you the dollar. [LAUGH] And then take it away? Or, the universe will take it away. Even if you don't have it. You'll have negative dollars. Oh, man. Okay, so Reward is very important here in a of couple ways. Reward is one of the things that, as I'm always harping on encompasses our domain knowledge. So, the Rewards you get from the state tells you the usefulness of entering into that state. Now. I wrote out three different definitions of r here, because sometimes it's very useful to think about them differently. I've been talking about the reward you get for entering into the state, but there's also a notion of reward that you get for entering into a state and taking an action. There's a rewar, or, being in a state and taking an action, there's a reward that you could get for being in a state, taking an action, and then ending up in another state as prime. It turns out these are all mathematically equivalent. But often it's easier to think about one form or the other. But for the purposes of the, you know, for the rest of this discussion, really you can just focus on that one, the reward of the value entering into a state. And those four things, by themselves, along with this Markov property and non-stationarity, defines what's called the Markov Decision Process. Or an MDP. Got it? I'm a little stuck on the, how those could be mathematically equivalent. Well we'll get to that later. Would you like a little bit of intuition? Sure. Well, you can imagine that just as before we were dealing with the the notion of making a non-Markovian thing Markovian by putting a little bit of history into your state. You can always fold in the action that you took to be in a state or the action that you took to get to a state, as a part of your state. But that would be a different Markov Decision Process. It would, but they would work out to have the same solution. Oh, I see.

I. Markov Decision Processes 4

Speaking of solutions, this is the last little bit of thing that you need to know. And that is. This defines a problem. But, what we also want to have, whenever we have a problem. Is a solution. So, the solution to the Markov Decision Process, is something called a policy. And, what a policy does. Is, it's a function, that takes in a state. And returns an action, in other words, for any given state that you're in, it tells you the action that you should take. Like as a hint? No, it just tells you, this is the at. Well, I mean, I suppose you don't have to do it, but the way we think about Markov Decision Processes, is that this is the action that will be taken. I see, so it's more of an order. Yes, it's a command. Okay. So that's all a policy is. A policy is solution to a Markov Decision Process. And there is a special policy, which I'm writing here as policy star, or the optimal policy, and that is the policy that maximizes your long-term expected reward. So if all the policies you could take, of all the decisions you might take, this is the policy that optimizes the amount of reward that you're going to receive or expect to receive over your lifetime. So, like, at the end? Well, at yeah, at the end, or at any given point in time, how much reward you're receiving. ¿From the Markov Decision Process point of view, there doesn't have to be an end. Okay. Though in this example, you don't get anything, and then at the end, you get paid off. Right, or unpaid off. Right. If you fall into the red square. So actually, your question points out something very important here. I mentioned earlier when I talked about the three kinds of learning that there, supervised learning and reinforced learning were sort of. Similar, except that instead of getting Ys and Xs we were given Ys and, Xs and Zs. And this is exactly what's happening here. Here what we would like to have if we wanted to learn a policy is a bunch of sa pairs as training examples. Well here's the state and the action you should've took, taken, here's another state and the action you should've taken, so on and so forth. And then we would learn a function, the policy, that maps states to actions. But what we actually see in the reinforcement learning world, in the Markov Decision Process world, is we see states, actions, and then the rewards that we received. And so in fact, this problem of seeing a sequence of states, actions, and rewards. It's very different from the problem of being told. This is the correct action to take to maximize a function. Or find a function that maps from state to action. Instead, we say well, if you're in this state, and you take this action, this is the reward that you would see. And then from that, we need to find the optimal action. So Pi star is being the F from that previous slide? Right. And R is being Z? Yes. And y is being a. And s is being x or x is being s Got you. Right. So but, I'm, okay I'm a little confused about this notion of a policy. So we have the, the, the thing we tried to do to get the goals was up, up, right, right, right. Yes. I don't see how to capture that as a policy. It's actually fairly straightforward. What a policy would say is: What state are you in? Tell me what action you should take. So,

the policy, basically is this: When you're in the state, start, the start state, the action you should take is up. And it would have a mapping. For every state that you might see, whether it's this state, this state, this state, this state, this state, this state, this state, or even these two states, and it will tell you what action you should take. And that's what a policy is. A policy, very simply, is nothing more than a function that tells you what action to take at every, in any state you happen to come across. Okay, but the, but the. The question that you asked before was about up, up, right, right, right. Mm hm. And, it seems like, because of the stochastic transitions. You might not be in the same state. Like, you don't know what state you're in, when you take those actions. No, so, one of the things for what we're talking about here, for the Markov Decision Process. Is, there're states, there're actions, there're rewards. You always know what state you're in, and you know what reward you receive. So does that mean you can't do up, up right right right? Well, the way it would work in a Markov Decision Process, so what you're describing is is what's often called a plan. You know, it's, tell me what sequence of actions I should take from here. What Markov Decision Process does and what a pr, a policy does is it doesn't tell you what sequence of actions to take from a particular state. It tells you what action to take in a particular state. You will then end up in another state because of the transition model, the transition function. And then when you're in that state you ask the policy what actions should I take now? Okay. Right, so this is actually a key point. Although we talked about it in the language of planning, which is very common for the people who di, for example take any ag course, the thing about this in terms of planning, what are the things that I can do to accomplish my goals? The Markov Decision Process way of thinking about it, the reinforcement way of thinking about it, or the typical reinforcement learning way of thinking about it, really doesn't talk about plans directly. But instead, talks about policies. Which from which you can infer a plan, but this has the advantage that it tells you what to do everywhere. And it's robust to the underlying stochastic of the word. World. So, is it clear that's all you need to be able to behave well. Well, it's certainly the case, that if you have a policy and that policy is optimal, it does tell you what to do, no matter what situation you're in. 'Kay. And so, if you have that, then that's definitely all you need to behave well. But I mean could it be that you wanted to do something like up, up, right, right, right which you cant write down as a policy? And why cant you write that down as a policy? Because the policies are only telling you what action to do as a function of the state not sort of like how far along you are in the sequence. Right unless, of course, you fold that into your state some how. But thats exactly right, the way to think about this is. The idea of coming up with a concrete plan of what to do for the next 20 time steps is different from the problem of whatever step I happen to be in, whatever state I happen to be in, what's the next best thing I can do? And just always asking that question. Hm. If you always ask that question, that will induce a sequence, but that sequence is actually dependent upon the set of states that you see. Whereas in the other case where we wrote down a particular policy, you'll notice that was only dependent upon the state you started in and it had to ignore the states that you saw along the way. And the only way to fix that would be to say, well, after I've taken an action, let me look at the state I'm in and see if I should do something different with it. But if you're going to do that, then why are you trying to compute the complete set of states? Or I'm sorry, the complete set of actions that you might take. Okay. Okay, so there you go. Now, a lot of what we're going to be talking about next Michael, is, given that we have MDP, we have this Markov Decision Process defined like this. How do we go from this problem definition to finding a good policy, and in particular, finding the optimal policy? That makes sense. Good. And there you go.

J. Sequences of Rewards 1

Alright. So having gone through that exercise, Michael, I think it's, it's worthwhile to step back a little bit and think about the assumptions that we've been making that have been mostly unspoken. And I'm going to say that the main assumption that we've been making in some sense boils down to a single word. And that word is stationary. So let me tell you what I mean by that and why by kind of illustrating what it is we've been sort of doing for a little while. Okay? Sure. Okay. So the first thing I'm going to say is that we've actually been. Kind of assuming infinite horizons. So what do I mean by that. When, when we think about the last grid world that we were playing with, we basically said well, you know, I want to avoid going to the end as quickly as possible if I have rewards of a certain value or whatever. Because, you know, the game doesn't end until I get to an absorbing state. Well, that sort of implies. That you basically can live forever. That you have an infinite time horizon to work with. Now can you, can you imagine why if you didn't have an infinite time horizon to work with you might end up doing something very different? Different then what, what we're doing in the grid world? Right, so here let me let me show you the game that we were, the grid world that we were doing before. Might help you think about it. So here's the grid world we had before. And as you recall. We had a particular policy that sort of made sense. Here, I'll, I'll write it out for you again. And this was with a case where we had a reward of minus 0.04. Remember? We just did this. Remember? Yep. Okay, and this was the policy that turned out to be optimal, and in the future I want you to pay attention to here is that when you're over right here near possible end state, rather than going up, it made sense to take the long way around. Because you're going to get some negative reward but it's a small enough negative reward compared to where you might end up. Okay with a positive one. Yeah, I see. And that makes some sense. Well, that only makes sense if you're going to be living long enough that you can take the long route around. What if I told you you only had say three times steps left, and then the game is going to end no matter where you end up? Well, it might be, it might make more sense to take some risk than just try to take the short way because there's really no chance you're going to get to the plus1. I'm entirely convinced of that though because there's still a chance you'll fall into the minus 1 along the way. Right, so the exact val, whether it makes sense to take the risk or not is going to depend upon two things, we've already talked about one of them which is the actual word that you get. If this reward were, you know, negative enough, then clearly it makes sense to just try to end things quickly, right? We just showed that in the last quiz. But another thing that it's going to depend upon is how much time you have in order to get to where you're going. If you've only got one or two time steps before everything's going to end. You can imagine that there are cases where, without changing the reward too much it makes a lot of sense to try to go ahead and quickly get to this plus 1, even though you have some chance of falling into the minus 1. As opposed to, trying to move

away, where you're then kind of, all but guaranteed that you're never going to reach the plus 1. So. Whether it makes sense to take the risk or not will depend upon the reward but it's also going to depend upon whether you have an infinite amount of time to get to where you want to get to or whether you have a finite amount of time. And the real major thing I want you to get out of that is that if you don't have an infinite horizon but you have a finite horizon then two things happen. One is the policy might change because things might end. But secondly, and more importantly, the policy can change, or will changee., even though you're in the same state. So, if I told you, if you're in this state right here, and I told you you didn't have an infinite amount of time, but you still had 100 million time steps then, it, I think it's clear that it still makes sense to go the long way around, right? Yeah, I mean the, the probability that this policy is going to last for a million timesteps has got to be tiny. Right. So, I might as well. It's 100 million timesteps might as well be infinity. But if I make that number not 100 million but I make it 2, or 3, or 4. Then suddenly your calculus might change. In particular, your caluculus will change even though I'm in the same state. Right? So maybe this state right here, if I've got a million, 100 million timesteps I still want got to go the long way around, but if I've only got a few time steps, the only way I'm ever going to get a positive reward is to go this way. Does that make sense? I guess so. So, you're saying, for example, even within the single run, it could be that I'm in a state and I try an action and maybe it doesn't work and I stay where I am. And I try it again, and maybe it doesn't work and I stay where I am. It might then switch to a different action, not because the other one wasn't working, but because now it's running out of time. Right, exactly. So we talked about this notion of a policy which maps states to actions, we talked about this notion about stationarity. So you believe that this sort of Markovian thing said, it doesn't matter where I've been it only matters where I am. And so if i'm in this state, since it only matters where I am, I'm always going to want to take the same action. Well that's only true in this kind of infinite horizon case. If you're in a finite horizon case. And that finite horizon, of course, is going to keep counting down, every time you take a step. Well then suddenly, depending upon the time step that's left, you might take a different action. So we could write that I think, just for the sake of kind of seeing it as some thing like, your policy is a function both the stature and, and the time step you're in. Hm. And that might lead you to a different set of actions. So this is important, this is important, I mean were not, we are not, for, for this course going to talk about this case at all, where you're in a finite horizon, but I think it's important for you to understand that the, without this infinite horizon assumption here. You loose this function of stationarity in your policies. Okay? Yeah. Interesting. Okay. So, that all, I think is, you know, making our something that's obvious, but becomes obvious after someone points it out to you. So, the second thing that I want to talk about, I think, is a little bit more subtle. And, and this notion of utility of sequences. So, as we've been talking, Michael, we have been sort of. Implicitly discussing not just the rewards we get in a single state, but the rewards that we get through a sequences of states that we take. And so I just want to point out a little fact that that comes from that, and where that ends up leading us. And then we'll get to some nice little cute series of math. So. Here's what I want to point what utilities, what we mean by utilities sequences. It means we have some function I'm going to call U for utility over the state, the sequence, sorry. Of states that we're going to see let's call them, S0, S1, S2 and so on and so forth. Well, I think an assumption that we've been making even if we haven't been very explicit about it is that if we had two sequences of states. S0, S1, S2, dot dot dot. And a different sequence S0, then S1 prime and S2 prime, that is two sequences that might differ from S1 on, but all start in the same start state. Okay? If we have a utility for the first, and that utility happens to be greater than the utility for the second, then it also turns out that we believe. That the utility for S1, S2, dot dot dot, is greater than the utility for S1 prime, S2 prime dot dot dot. Alright so these are two different sequences, S one, the S's and the S prime's are two different sequences. Yes. And in the beginning we're comparing them with S0 stuck in front of both of them. And we're saying if I prefer the S0 followed by all the S's, to S0 followed by the S primes, then I have that same preference even with those S0s missing. Right, and so this is called stationarity of preferences. And, another way of saying it is. That if I prefer one sequence of states today over another sequence of states, then I prefer that sequence of states over the same sequence of states tomorrow. So isn't, isn't this just obvious? Because the whatever the rewards for those two cases, we're just adding the reward we get for S0. So. It's going to be the same. But listen to what you just said. You just said, well, it'll be the same, because all we're doing is adding the reward for S0. But what did we ever say about adding up rewards? I thought, I thought that's what we were doing. That's right, that is what we were doing. But we never actually sat down and wrote that down and said, this is what it means. To talk about the utility of a sequence of states as opposed to the reward that you get in one state. Okay, so you're saying that if we, if we are adding rewards, then this follows. Right. Okay. And then I've actually been saying something even stronger, which is, I will show you on the next slide, which is if you believe that this is true, that the utility of one sequence of states is greater than the utility of another sequence of states. Both today and tomorrow. Then it actually forces you to do some variation of what you said which is just adding sequences of states. Or adding the rewards of the sequence of states that we see. That's really interesting. So then, so the adding isn't really an arbitrary thing it follows from this, this deeper assumption. Right, and the reason I bring this up is because. It would make sense if you were to just to grab someone off the street and start talking about Marco Decision Processes. One of two things will happened. Either they'd run screaming from you like you're a crazy person or they would sit and they would listen and if they listen they would just completely buy into the idea that you just add up sequences of rewards. You know, sequences of rewards that you see as a way of talking about how good the states are because that's a very natural thing to do. But it turns out that mathematicall if you have this notion. A sort of stationary of preferences and this sort of infinite arise in world. You really are in a case where this has to be true. And it has to be the case if you have to do some form of addition. Because nothing else sort of can be guaranteed to maintain this property over stationary preferences. I mean, as you said, if I got one sequences of states and another sequnce of states and by just prepending or appending another set of states to it, I'm still going to always guarantee that one's greater than the other. You kind of have to do some form of adding the reward that you see in the states in both cases. because if you don't do that, then eventually this inequality will not hold. So, let me write that down in math terms. And see where that gets us, okay? Cool.

K. Sequences of Rewards 2

Alright, Michael. So, let's write that down in math, okay? You like math, right? I do. Okay. So here's the math version of it. I'm going to just say that the utility that we receive for visiting a sequence of states, S0, S1, S2 ellipsis, is simply the sum of all the rewards that we will receive for visiting those states. Sure. Does that make sense? Yeah. Is that consistent with what you were doing when you were thinking about the grid world? Yeah, exactly. But I thought we were going to make that not definitional, we were going to derive that it had to be that way. No, we're not. They can read about it. It'd take me, like, an hour and a half to do it. Alright. And I'm already losing my voice. But. What I do want people, I want you to believe is that the utility of the sequence of states, thinking of it as the sum of all the rewards, makes sense, right? And then if nothing else, at least it makes, it's consistent with what you've been doing as we've been talking about this grid work. Yeah, absolutely. I mean it, it also kind of makes me think about money. Go on. Well, so, so, I mean, that's sort of how money works. If we get some kind of payoff each day, those payoffs get added to each other. They don't get, you know, subtracted or multiplied or square rooted or whatever. They just, you know, they go into your bank account, and things add. So this feels like, kind of like that. It's like money in my pocket. Sure, if you have a big enough pocket. Okay, I'm with you on that. Alright, so I'm going to say that this all makes sense. It, it's really awesome and it sort of doesn't work. And to illustrate why this doesn't work, I'm going to give you a quiz. Yay. because you like quizzes. So I've been told.

L. Sequences of Rewards 3 Question

So, here's the quiz. You see on the screen this sort of unexplained two little squiggles with a bunch of numbers in front, on top of them or under them or near them? Yeah I, I assume that's a river and on one bank it's the land of the plus ones and the other bank has been infilitrated by plus twos. That's not what I intended at all but I actually like that enough that I'm going to pretend, that that's what I intended. So these work out to be sequences of states. Oh, I see now. Okay, and rewards that you receive. No, no, no, it's a riverbank and on one side of the bank, as you walk along it, you get a bunch of plus ones. On the other side you get a bunch of plus ones and some plus twos. And this just goes on forever, okay? And, I'm not going to tell you what all the rewards are after that except that they, they look similar to the rewards that you see here. Okay. Plus ones and plus twos, okay? And let's say the top, the top one, in fact, nothing but plus ones. And the bottom ones are some plus ones with maybe some plus twos sprinkled in and out, but those are really the only options. Okay? Yeah. Here's my question to you. So, would you rather be on the top side of the river bank, or the bottom side of the river bank? Okay? You think you know the answer already, don't you. Yeah. Okay, cool. Well then, let's, let's give our listeners a, a chance to come up with the right answer.

M. More About Rewards 3 Solution

Okay, Mikey, you got answers for me? sure. Alright. Wait, should I, do need to make any assumption, or can I make any assumptions I want about the non-blue states? No. It doesn't matter. [LAUGH] Actually. [LAUGH] So the answer is no. Maybe it does. The answer is no. But it doesn't actually matter because Remember, it's all very stationary and Marcovian. So the only thing that matters is where you are. You're stationary Marcovian. Alright, so I think, okay. So let's, let's do the first one, then. Okay. So, so this is, this is pretty cool. So by changing the reward the way that you did, it is now, it's not a hot beach anymore. It's like. You know like, super awesome money beach. Yes, it's like a beach made of bacon. Sure. Okay. Let's think of it that way. Although neither of us is eating bacon at the moment. At the moment. But. [CROSSTALK] Great, because, because then our mouths would be full and it would be very difficult to give this lecture. Alright so, no, no, no,no, no I mean like, even. Alright, never mind. The point is that it's awesome, and there's like diamonds. I don't know, I just saw The Hobbit movie, and there is a room filled entirely with treasures. And so, this is kind of like that. And so now the question is, what should I be doing, like I should never go to the plus 1 or the minus 1, because that ends my treasure gathering. I should just continue to stay in states that aren't those states. Okay. So I would say left for the, for the top state. Mm-hm. Let's say left for the bottom state. This one? The, yes, uh-huh. Okay. And the other two, I need to think about for a moment. So I feel like I'd like to get away from the minus 1. Mm-hm. So, like, you know, to go up. But then that would give me some chance of slipping, wouldn't it? Mm-hm. let's see, I don't want to go to the right. I don't want to go down. I don't want to go. I'm going to go to the left. Mm-hm. I'm going to bash my head against the wall, because then I get money. Yeah. That's weird. There's really no pain for running into the wall? No. You just stay where you are. Alright and for the bottom one, I feel like it's the same kind of thing. I don't want to go to the right or to the left, cause then there's a probability that I'll slip and end up ending my game. So I'm going to go down. yeah. I like that. So most of these are right, and one of them is wrong. Or, it's not that it's wrong, it's just that's not as right as it could be. Which one do you think that is? I think they're right. But if, if there's one that I could imagine you not being so happy with, which is the bottom one, There are two bottom ones. You mean this one? Yeah. Yeah. What would I like? I think you might prefer if I bash my head against the wall. No, actually, it turns out that this is a correct answer so all four of these are correct. But in this particular state, it actually doesn't matter which way you go. Oh, I see. A more complete answer. Yes. Would be, doesn't matter. Yeah. And is that true of any of the other ones? No. I guess, I guess you're right. The other ones, you have to go in the direction indicated. Right. So, by the way, just like here, it doesn't matter which direction you go in. It's actually true for all of the other states as well. It doesn't matter where you go because In these three states, these are the only states where you can accidentally, you could end the game. And in each of them, you can take an action that guarantees that you don't end the game. So it really doesn't matter about where you do it in the other states. Cool. Right? So that's a good argument. This makes a lot of sense. So at the end of the day, basically, because the reward is positive. And you're just accumulating reward or diamonds and treasure, as you put it. You just never want to leave, so you always want to avoid either of these states. It doesn't matter that there's a +1 here and a -1 here. If I can stay here forever, then I just accumulate positive reward forever. Okay, so what's the next one?

So, interesting. So now, now the beach is really, really hot. It's minus 2, so. Mm-hm. I want to get off here as quickly as I can. So definitely the top one, I would go to the right. Just try to get, get that +1 and get out of, get outta here. Yeah. Alright, so then, let's think about the bottom right. Okay. So, in the best of all possible worlds I go around and dump into the +1. And that would take me again if, if I don't slip at all one, two, three, four steps to do that Mm-hm. Actually wait one, two, three, three steps one the hot sand. Yup And I'm getting and -2 for each of those. So that's like minus 6. Mm-hm. So I could get that plus 1 at the end, and that makes it only minus 5. But I think I'd rather just get off the beach. So I'm going to say in the bottom right to actually dive into the pit of pain, because it can't be as bad as where I am. That's exactly right. So, okay, so then the, the one next to it on the left is a little trickier. If I go up to the plus 1, I get a minus 2 Followed by a +1 which is also -1. Mm-hm. So okay. I'm not so sure about that one. Well remember you'll always have some probablitly of ending up back where you started Is that true? Oh because, if I. Well no, no, that's not true if I dive straight into the -1. Then I have 0.8 chance of going to the -1. 0.2 of going up. To the arrow that's labeled, 0.2 going down to the blue box that's not labeled Mm-hm. Wait, 0.2? No, 0.1 Mm-hm. Whereas if I go up, right, I might stay where I am. And accrue -2, like, yeah I feel like I need a calculator for this. Yeah, but you're, I think your intuition is right, though. Just think about it as How much, I mean what do you think has the best chance of most quickly ending things? Well certainly jumping into the -1. Right. I think that's going to be marginally better because there's only one chance of slippage and delay. Where as if I take additional step, there's two chances of slippage and delay. Exactly, that's exactly the right kind of argument to make and, and because the -2 is so big. You're you're going to, you're just better off ending it even if you ended in pain, in the same way that you were in the bottom right hand corner. And so bottom left I would say the cleanest thing would be to jump into the -1. So to go to the right so that I can go up and get into the -1. And what's nice about that move is that is that you either end up in the bottom right hand corner. Or you end up staying where you are, or you end up moving up. But in either case, you are always sort of, you know, you never get farther away, right? Whereas if you try to go up, there's a chance that you will get farther away. Interesting. Is there a way to be sure about these? Yeah, you just have to do the math. [LAUGH] Do we know how to do the math? Yeah, you just start, you just do expectations. You can basically just say, well what's the expected time. Since you know what you're trying to do here is to get to the end. You can just, you can calculate the expected length of time it'll take you to get there. And then just, you know, multiply by all the rewards you're going to get in the meantime. If it helps, though, I will tell you that for any value where the reward is less than minus 1.6284. [LAUGH] This is the right thing to do. [LAUGH] Okay. Just so you know. Alright, so you're saying you actually do know that this is the optimum. The thing that we have here. Yes, I know that this is the optimum. Alright. Let me just draw out the rest of it for you. So it's, it's I think interesting to compare this one. The bottom right one, where you have given me negative rewards to encourage me to end the game. With our main one up here in the upper left. Where you have also given me some negative reward to encourage me to end the game. If you look carefully you'll notice that in this one, you're encouraged to end the game. But you're really encouraged to end the game at a plus 1. Here the reward is so strongly negative, you just need to end the game. And so you end up with a different response, here. Because this is, is just as quick to get to the minus 1 and into pain as to, take this extra step and try to get to the plus 1. And you end up with a different response here. Here, here, and here. That's really interesting. Yeah, it is. And so, these changes matter. They matter a great deal. So, how am I suppose to choose my rewards? Carefully and with some forethought. Nice. But, you know, again There's lots of ways to think about this. So the way we've been sort of talking about MDPs is kind of the way we've been talking about we talked in the first part of the course. When we talked about having teachers and learners. You can think of your reward as sort of the teaching signal. Sort of the teacher telling you what you ought to do, and what you ought not to do. But another way of thinking about this is that because the rewards define the MDP The rewards are our domain knowledge. I see. So if you're going to design an MDP to capture some world. Then you want to think carefully about how you set the rewards in order to get the behavior that you wish. That's fair. I mean, it seems a little bit like a cop out, but I think it seems like a necessary one. Yeah. I mean, there's really, I mean, again. No matter what you do, you've gotta be able to inject domain knowledge somehow. Otherwise, there's no learning to do. And in this case, the reward basaiclly is telling you how important it is to get to the end. Cool. Okay. Alright. Let's move on.

N. Sequences of Rewards 1

Alright. So having gone through that exercise, Michael, I think it's, it's worthwhile to step back a little bit and think about the assumptions that we've been making that have been mostly unspoken. And I'm going to say that the main assumption that we've been making in some sense boils down to a single word. And that word is stationary. So let me tell you what I mean by that and why by kind of illustrating what it is we've been sort of doing for a little while. Okay? Sure. Okay. So the first thing I'm going to say is that we've actually been. Kind of assuming infinite horizons. So what do I mean by that. When, when we think about the last grid world that we were playing with, we basically said well, you know, I want to avoid going to the end as quickly as possible if I have rewards of a certain value or whatever. Because, you know, the game doesn't end until I get to an absorbing state. Well, that sort of implies. That you basically can live forever. That you have an infinite time horizon to work with. Now can you, can you imagine why if you didn't have an infinite time horizon to work with you might end up doing something very different? Different then what, what we're doing in the grid world? Right, so here let me let me show you the game that we were, the grid world that we were doing before. Might help you think about it. So here's the grid world we had before. And as you recall. We had a particular policy that sort of made sense. Here, I'll, I'll write it out for you again. And this was with a case where we had a reward of minus 0.04. Remember? We just did this. Remember? Yep. Okay, and this was the policy that turned out to be optimal, and in the future I want you to pay attention to here is that when you're over right here near possible end state, rather than going up, it made sense to take the long way around. Because you're going to get some negative reward but it's a small enough negative reward compared to where you might end up. Okay with a positive one. Yeah, I see. And that makes some sense. Well, that only makes sense if you're going to be living long enough that you can take the long route around. What if I told you you only had say three times steps left, and

then the game is going to end no matter where you end up? Well, it might be, it might make more sense to take some risk than just try to take the short way because there's really no chance you're going to get to the plus1. I'm entirely convinced of that though because there's still a chance you'll fall into the minus 1 along the way. Right, so the exact val, whether it makes sense to take the risk or not is going to depend upon two things, we've already talked about one of them which is the actual word that you get. If this reward were, you know, negative enough, then clearly it makes sense to just try to end things quickly, right? We just showed that in the last quiz. But another thing that it's going to depend upon is how much time you have in order to get to where you're going. If you've only got one or two time steps before everything's going to end. You can imagine that there are cases where, without changing the reward too much it makes a lot of sense to try to go ahead and quickly get to this plus 1, even though you have some chance of falling into the minus 1. As opposed to, trying to move away, where you're then kind of, all but guaranteed that you're never going to reach the plus 1. So. Whether it makes sense to take the risk or not will depend upon the reward but it's also going to depend upon whether you have an infinite amount of time to get to where you want to get to or whether you have a finite amount of time. And the real major thing I want you to get out of that is that if you don't have an infinite horizon but you have a finite horizon then two things happen. One is the policy might change because things might end. But secondly, and more importantly, the policy can change, or will changee., even though you're in the same state. So, if I told you, if you're in this state right here, and I told you you didn't have an infinite amount of time, but you still had 100 million time steps then, it, I think it's clear that it still makes sense to go the long way around, right? Yeah, I mean the, the probability that this policy is going to last for a million timesteps has got to be tiny. Right. So, I might as well. It's 100 million timesteps might as well be infinity. But if I make that number not 100 million but I make it 2, or 3, or 4. Then suddenly your calculus might change. In particular, your caluculus will change even though I'm in the same state. Right? So maybe this state right here, if I've got a million, 100 million timesteps I still want got to go the long way around, but if I've only got a few time steps, the only way I'm ever going to get a positive reward is to go this way. Does that make sense? I guess so. So, you're saying, for example, even within the single run, it could be that I'm in a state and I try an action and maybe it doesn't work and I stay where I am. And I try it again, and maybe it doesn't work and I stay where I am. It might then switch to a different action, not because the other one wasn't working, but because now it's running out of time. Right, exactly. So we talked about this notion of a policy which maps states to actions, we talked about this notion about stationarity. So you believe that this sort of Markovian thing said, it doesn't matter where I've been it only matters where I am. And so if i'm in this state, since it only matters where I am, I'm always going to want to take the same action. Well that's only true in this kind of infinite horizon case. If you're in a finite horizon case. And that finite horizon, of course, is going to keep counting down, every time you take a step. Well then suddenly, depending upon the time step that's left, you might take a different action. So we could write that I think, just for the sake of kind of seeing it as some thing like, your policy is a function both the stature and, and the time step you're in. Hm. And that might lead you to a different set of actions. So this is important, this is important, I mean were not, we are not, for, for this course going to talk about this case at all, where you're in a finite horizon, but I think it's important for you to understand that the, without this infinite horizon assumption here. You loose this function of stationarity in your policies. Okay? Yeah. Interesting. Okay. So, that all, I think is, you know, making our something that's obvious, but becomes obvious after someone points it out to you. So, the second thing that I want to talk about, I think, is a little bit more subtle. And, and this notion of utility of sequences. So, as we've been talking, Michael, we have been sort of. Implicitly discussing not just the rewards we get in a single state, but the rewards that we get through a sequences of states that we take. And so I just want to point out a little fact that that comes from that, and where that ends up leading us. And then we'll get to some nice little cute series of math. So. Here's what I want to point what utilities, what we mean by utilities sequences. It means we have some function I'm going to call U for utility over the state, the sequence, sorry. Of states that we're going to see let's call them, S0, S1, S2 and so on and so forth. Well, I think an assumption that we've been making even if we haven't been very explicit about it is that if we had two sequences of states. S0, S1, S2, dot dot dot. And a different sequence S0, then S1 prime and S2 prime, that is two sequences that might differ from S1 on, but all start in the same start state. Okay? If we have a utility for the first, and that utility happens to be greater than the utility for the second, then it also turns out that we believe. That the utility for S1, S2, dot dot dot, is greater than the utility for S1 prime, S2 prime dot dot dot. Alright so these are two different sequences, S one, the S's and the S prime's are two different sequences. Yes. And in the beginning we're comparing them with S0 stuck in front of both of them. And we're saying if I prefer the S0 followed by all the S's, to S0 followed by the S primes, then I have that same preference even with those S0s missing. Right, and so this is called stationarity of preferences. And, another way of saying it is. That if I prefer one sequence of states today over another sequence of states, then I prefer that sequence of states over the same sequence of states tomorrow. So isn't, isn't this just obvious? Because the whatever the rewards for those two cases, we're just adding the reward we get for S0. So. It's going to be the same. But listen to what you just said. You just said, well, it'll be the same, because all we're doing is adding the reward for S0. But what did we ever say about adding up rewards? I thought, I thought that's what we were doing. That's right, that is what we were doing. But we never actually sat down and wrote that down and said, this is what it means. To talk about the utility of a sequence of states as opposed to the reward that you get in one state. Okay, so you're saying that if we, if we are adding rewards, then this follows. Right. Okay. And then I've actually been saying something even stronger, which is, I will show you on the next slide, which is if you believe that this is true, that the utility of one sequence of states is greater than the utility of another sequence of states. Both today and tomorrow. Then it actually forces you to do some variation of what you said which is just adding sequences of states. Or adding the rewards of the sequence of states that we see. That's really interesting. So then, so the adding isn't really an arbitrary thing it follows from this, this deeper assumption. Right, and the reason I bring this up is because. It would make sense if you were to just to grab someone off the street and start talking about Marco Decision Processes. One of two things will happened. Either they'd run screaming from you like you're a crazy person or they would sit and they would listen and if they listen they would just completely buy into the idea that you just add up sequences of rewards. You know, sequences of rewards that you see as a

way of talking about how good the states are because that's a very natural thing to do. But it turns out that mathematicall if you have this notion. A sort of stationary of preferences and this sort of infinite arise in world. You really are in a case where this has to be true. And it has to be the case if you have to do some form of addition. Because nothing else sort of can be guaranteed to maintain this property over stationary preferences. I mean, as you said, if I got one sequences of states and another sequnce of states and by just prepending or appending another set of states to it, I'm still going to always guarantee that one's greater than the other. You kind of have to do some form of adding the reward that you see in the states in both cases. because if you don't do that, then eventually this inequality will not hold. So, let me write that down in math terms. And see where that gets us, okay? Cool.

O. Sequences of Rewards 2

Alright, Michael. So, let's write that down in math, okay? You like math, right? I do. Okay. So here's the math version of it. I'm going to just say that the utility that we receive for visiting a sequence of states, S0, S1, S2 ellipsis, is simply the sum of all the rewards that we will receive for visiting those states. Sure. Does that make sense? Yeah. Is that consistent with what you were doing when you were thinking about the grid world? Yeah, exactly. But I thought we were going to make that not definitional, we were going to derive that it had to be that way. No, we're not. They can read about it. It'd take me, like, an hour and a half to do it. Alright. And I'm already losing my voice. But. What I do want people, I want you to believe is that the utility of the sequence of states, thinking of it as the sum of all the rewards, makes sense, right? And then if nothing else, at least it makes, it's consistent with what you've been doing as we've been talking about this grid work. Yeah, absolutely. I mean it, it also kind of makes me think about money. Go on. Well, so, so, I mean, that's sort of how money works. If we get some kind of payoff each day, those payoffs get added to each other. They don't get, you know, subtracted or multiplied or square rooted or whatever. They just, you know, they go into your bank account, and things add. So this feels like, kind of like that. It's like money in my pocket. Sure, if you have a big enough pocket. Okay, I'm with you on that. Alright, so I'm going to say that this all makes sense. It, it's really awesome and it sort of doesn't work. And to illustrate why this doesn't work, I'm going to give you a quiz. Yay. because you like quizzes. So I've been told.

P. Sequences of Rewards 3 Question

So, here's the quiz. You see on the screen this sort of unexplained two little squiggles with a bunch of numbers in front, on top of them or under them or near them? Yeah I, I assume that's a river and on one bank it's the land of the plus ones and the other bank has been infilitrated by plus twos. That's not what I intended at all but I actually like that enough that I'm going to pretend, that that's what I intended. So these work out to be sequences of states. Oh, I see now. Okay, and rewards that you receive. No, no, no, it's a riverbank and on one side of the bank, as you walk along it, you get a bunch of plus ones. On the other side you get a bunch of plus ones and some plus twos. And this just goes on forever, okay? And, I'm not going to tell you what all the rewards are after that except that they, they look similar to the rewards that you see here. Okay. Plus ones and plus twos, okay? And let's say the top, the top one, in fact, nothing but plus ones. And the bottom ones are some plus ones with maybe some plus twos sprinkled in and out, but those are really the only options. Okay? Yeah. Here's my question to you. So, would you rather be on the top side of the river bank, or the bottom side of the river bank? Okay? You think you know the answer already, don't you. Yeah. Okay, cool. Well then, let's, let's give our listeners a, a chance to come up with the right answer.

Q. Sequences of Rewards 3 Solution

Okay Michael, I didn't give you as much time as I normally do because you think you already know the answer. So what's the answer? So, you know, I'd rather get the plus 2's occasionally, so I'll say the bottom one. No. Wait, what? You're not going to tell me the top one's better? no, it's not. Okay then that feels like a trick question. It's not a trick question. The answer is, neither one is better than the other. Oh, so I had to give neither? Or both. Which is better, and I can click on neither. Or you could click on both. Okay. So you thought that the bottom one was better, what was your reasoning for that? Because, sort of moment to moment, sometimes I'm getting plus one's, and that matches what I would've gotten if I had been on the other bank. But then sometimes I get plus two's, which is actually better than what I would have gotten on the other bank. And so, I'm, I'm, I never feel any regret being on the bottom bank. I only feel regret being on the top bank. That's fine. So, what would you say the utility of the sequence along the bottom actually is? 1 plus 1 plus 2 plus 1 plus 1 plus 2 plus dot, dot, dot. Which is equal to? infinity, I guess. That's right. What about the utility of the top one? 1 plus 1 plus 1 plus, that's also infinity. Yep. So the utility of both of those sequences is equal to infinity. Do you think one of them is bigger than the other? You drew the bottom one a larg, a little bit larger. I did. Or longer, anyway. Yeah. But, in the end, the sum of the rewards that you get are both going to be infinity. So the truth is, neither one of them is better than the other. I still don't think you can say that both are better. You can't get around that. But yeah, I see, I see, neither, I can see neither is better. Or that both are better. Neither is better, both is better, whatever. The point is that, they're both equal to infinity. Hm. And the reason they're equal to infinity is because all we're doing is accumulating rewards. And, if we're always going to be able to get positive rewards no matter what we do, then it doesn't matter what we do. This is the existential dilemma of being immortal. Oh, living forever. Right. So if you live forever then, like why should you care about anything ever? Right I mean, everyone, every all the mortal people are going to die and one day they'll all be, you know, an infinite amount of time in your past. I could do this thing here, which is pleasurable, or I could do this thing right now, that, you know, will, is less pleasurable, but will eventually get me to a better place. But if I'm going to live forever, and I can always get to a better place, than it really doesn't matter what I do. It really doesn't matter. Mm. Because I'm just accumulating rewards, I'm living forever, and I'm

going to, infinity is infinity and there's no really no way to compare them. Having said that, your original notion that, look, it feels like I should never regret having taken the second path compared to the first because I will occasionally do better. Seems like the right intuition to have. I see but it's just not built into this particular utility scheme. Right, but it turns out there's a very easy way we can build it into this utility scheme by just making one tiny little change. Would you like to see it? Yes. Beautiful, let's see it then

R. Sequences of Rewards 4

Okay Michael so here's the little trick. So all I've done is I have replaces the equation at the top with an alternate version of the equation. 'Kay. So it looks there's, you're now exponentially blowing up the reward. I am not exponentially blowing up the reward. Instead of what I've done. Is I've added of the see, the rewards that I'm going to see. For the states that I'm going to see. And I multiplied by gamma to the T. The gamma is between 0 and 1. I see. So... Do you? Well, kind of. So, so it doesn't exponentially blow up. It exponencially implodes. Right. So, so things that are like a thousand steps in the future If we multiply, if we take something that's less than 1 and we raise it to that power, it goes essentially to 0. So it's like it starts off the rewards are kind of substantial and then they get, they trail off quickly. Right. And in fact we can write down mathmatically what this is If you actually, if you stare at it long enough and you remembered your intermediate algebra or your calculus or wherever it is you learned this stuff. You would probably recognize this as a special kind of sequence or series. Do you remember what it is? Television series? [LAUGH]. No, but that's remarkably close. So let's see if we can bound this particular equation. So we know that this version of the equation eventually ends up being infinity if all the rewards are positive. Right? Mm, hm-mm. So what does this end up being even in the case where all the rewards are positive? Well, we can bound this from above by the largest reward that we will ever see, in the following way. So all I've then is said well I don't know what the rewards are but there is some maximum reward I'm going to call it Rmax. And if I pretended that I always got the Rmax reward. So long as that, as long as that's an actual number. A finite number. I know that this is bounded from above by this expression. Well what does this look like. Maybe this does look like a series you remember. Geometric series? Yes. This is the geometric series. And this is exactly equal to this. . And I assume that the summation causes the max to become lower case. Yes, yes it does. That's a, that's a trick that they don't really go over deeply in calculus, but it's true. Oh, that's cool. Alright so, the reward, the maximum reward divided by 1 minus gamma. So when gamma is really close to 0, you're just getting, oh you're getting just that one reward in the beginning and then everything falls off to nothing after that. Yep. And if gamma is really close to 1. You're dividing by something that's teeny tiny, which actually is like multiplying it by something that's really big. So it kind of magnifies the reward out, but it's not infinity until gamma get's all the way up to 1 and that's the same case we were in before. Right, so in fact you'll notice the way I wrote this, gamma has to be between 0 and 1, 0 inclusive but strictly less than 1. If I made it so that it could include 1, well that's actually the first case. Got it. Because 1 to a power is always 1, and so, this is actually a generalization of the sort of, infinite sum of rewards. This is called discounted rewards, or discounted series, or discounted sums, and it allows us, it turns out, to go an infinite distance in finite time. Wait. That makes sense. No. No. [LAUGH] An infinite distance? Yeah. That's at least the way I like to think about it. So, if we're discounted in this particular way, then that gives us a geometric series. And what it does is it allows us to add an infinite number of numbers, but it actually gives us a finite number. Cool. So this means we can still have our, and it turns out, by the way, just to be clear, that this world that we're in. Where we are in between 0 and 1, 0 inclusive is still consistent with our infinite horizons and our stationarity over preferences. So we can add things together and be consistent with those assumptions we were making before Or we can do discounted rewards and we're still going to be consistent with what we're doing before. But the difference between this case and this case is that by doing the discount, we get to add infinite sequences and still get something finite. The intuition here is that Since, if gamma's less than 1, then eventually as you raise it to a power, it will basically become 0. I mean, that's why it's a geometric series. Which is like having a finite horizon. But that horizon is always the same finite distance away, no matter what, where you are in time. Which means it's effectively infinite. Or unbounded, or however you want to think about it. So it's like you take a step, but you're no closer than where, when you started. Right, so, what does that mean in, a world where you meant to say you're no closer than where you were trying to end up? So that means that even though, you've taken a step forward, the horizon sort of remains a fixed distance away from where you are. Mmm, mmhmm. That's what it means to To, to have this kind of this gamma value here and so you can still treat it as if it's always going to be infinite, even though it gives you a finite value, and that gives you your stationary. So does that make sense? I mean that's sort of the intuition. Yeah. You really are have an in, you're always in infinite, It's always in, infinity. But the gamma value means that at any given point in time, you only really have to think about what amounts to a finite distance. You never get closer to the horizon, even though you're always taking steps towards it. But there still is a horizon that you can see, and I can kill that analogy. [LAUGH] So can we, can you explain to me why it has this form, this R max over 1 minus gamma? I could. There's a You can kind of prove that, in a little cute math way, and I'm happy to take that diversion if you want. Yeah, I think so. I mean, unless, unless you were going to tell me something else. Well how about I tell you something else and then I take that diversion? Okay. Okay, so here's the something else. In the beginning I said, well, this is like going in infinite amount of distance in finite time, which you rebelled against. And my explanation is more about. How infinity looks finite which is not the same thing as going an infinite amount of distance in finite time. The reason I said that is beacuse of the sningularity. What? The singularity so, some of our listeners no doubt have heard of the singularity. You never heard of the singularity? The singularity is like when computers get so fast that you do infinite computation and, oh. Right. So the singularity, for those of you who don't know is is this sort of observation that has been made by many folks. That the sort of limit to computer power growing faster is the fact that it takes us amount, some amount of time to design the next generation of computers. Right, So Moore's law says everything doubles you know, everything 6, 18 months or whatever. But if we could design things faster, then we could actually make it double more quickly. So, one day we'll get to the point where the computer, which I will draw here, like that. The computer can actually design the next generation of

computer. Well, when it designs the next generation of computer, that computer will also have the ability to design the next generation of computer. But it will be able to design it twice as fast. And then the next one will be able to design its next successor twice as fast, and so on and so forth. So this little time between generations, will keep Having every single time, which looks remarkably like a geometric sequence. And so once we reach this point, where the computer can actually design its successor, we will then be able to do an infinite number of successors. In finite time. [LAUGH] And that's when the world comes to and end. And that's what's called the singularity. Because we can't understand what happens after that point. So that's what I mean by going an infinite amount of distance in finite time. It just doesn't seem like distance. But, yeah, that's a weird example, for sure. I think it makes sense. [INAUDIBLE] infinite number of doublings in it. Yeah, [INAUDIBLE], no. [LAUGH] Well, we'll, we'll, we'll do we'll do some assignment where we allow people to decide whether this makes sense or one of your analogies make sense. And I'll just remind the listener that I'm the one who'll be assigning final grades. Okay. So with that let's go and do your little diversion. And and then we'll come back to. Doing a whole lot of math. So I actually think this little nice diversion will be warmup to all the math we're going to be doing. Okay? Oooh. Hm. More math. Hm.

S. Assumptions

So, if we think about the equation that we were doing before, which was you know, the sum of all these gammas times some R max. We can basically take out the R max and what we end up with is something that looks like that, right? You're summing together a bunch of gammas and then multiplying the result by R max. So what does that acutally look like? Well that looks like this. Gamma to the zero, plus gamma to the one, plus gamma squared, plus dot dot dot dot. Eventually multiplied by R max. Right? So let's call this x, okay? Does x include the R max or not? No. So we'll just call the little sequence of, it's going to turn out not to matter. Let's just call the sequence of gammas we're adding together, let's just call that x. Good. Well, if you look at it, this is actually recursive. Right? Because this is an infinite sequence, if you sort of shifted it one over in time, you would end up with just the repeated sequence again. All right? Which means that we can write x in terms of itself. x equals gamma zero plus gamma times X. I see, so it's shifted over, but then you have to multiply it again by gamma, to get up to gamma one, gamma two. Right, exactly. So, so, this is just, you know, it's just math, that's just all it is. so, we can try to solve for x and figure out what x is, right? Cool. So, what is x? Well, we can subtract from both sides and so we end up with like, something like x minus gamma x equals gamma 0. Right? Seems like we can stop writing gamma 0. Isn't that just one? Yes, but I've already, I've gone too far, Michael. I've alrea, I've already written gamma 0. Alright. [LAUGH] so, this becomes x times 1 minus gamma equals gamma 0, or as Michael so astutely points out, 1. [LAUGH]. Alright. Which means what? It means that. So then we divide by 1 minus gamma. a is 1 over 1 minus gamma. Neat. And that, of course, we are going to multiply by R max. To get the formula that we had. That's, yeah, that's nifty. Yeah, there we go. So, why do we do this? Well, because you like stuff like this and it points out something very simple, which is that geometry is easy. [LAUGH] I don't think that's the kind of geometry that people usually think of. Well, they should be thinking of this kind of geometry. So geometry is easy. And by doing a little cute algebra, we can derive ridiculous equations that turn out to help us deal with go an infinite distance in finite time. [LAUGH] I don't think that's what they're doing, but okay. [LAUGH] So, let's think of this as warmup for what I actually wanted to show you, which is going to turn out to be a whole lot of math. Okay? Alright, let's, let's go for it. Alright, let's do that.

T. Policies 1

Okay, so Michael, in the spirit of what we just went through in deriving the geometric series, I'm now going to write down a bunch of math. And what I'm going to do is I'm just going to say it at you, and you're going to interrupt me if it doesn't make sense. Okay? That makes sense. It does. Okay, so here's the first thing I'm going to write down. I've decided that by going through this excercise of Of utilities in this kind of reward, we can now write down what the optimal policy is. The optimal policy, which as you recall is simply pi star, is simply the one that maximizes our long-term expected reward. Which looks like what? Well, it looks like this. There, does that make sense? Let me think, so. We have an expected value of the sum, of the discounted rewards, at time t. And, given, pi. Meaning that we're going to be following pi? Mm-hm. So these are the, the sequence of states we're going to see in a world where we follow pi. And it's an expectation because, things are non-deterministic. Or may be non-deterministic. And do we know which state we started? It doesn't matter, it's whatever s zero is. I see. Whatever s zero is, but isn't that random? I mean s one and s two, s three; those are all random. Well, we start at some state, it doesn't matter, so t is starting out a zero. And going to infinity. Okay? So does this make sense? Yes, so then, so we're saying, I would like to know the policy that maximizes the value of that expression. So it gives us the highest expected reward. Yeah, that's the kind of policy I would want. Exactly. So, good, we're done, we know what the optimum policy is. Except that it's not really clear what to do with this. All we've really done is written down what we knew it was we were trying to solve. But it turns out that we've defined utility in such a way that it's going to help us to solve this. So let me write that down as well. I'm going to say that the utility of a particular seque, of a particular state okay. Well it's going to depend upon the policy that were following. So I'm going to rewrite the utility that takes the superscript pie. And thats simply going to be the expected set of states that I'm going to see from that point on given that I've followed the policy. There, does that make sense? It feels like the same thing. I guess the difference now is that you're saying the utility of the policy out of state is what happens if we start running from that state. Yep. And we follow that policy. Got it. Right. So, this answers the question you asked me before about, well, what's S0? Well, we talk about that in terms of the utility of the state. So how good is it to be in some state? Well, it's exactly as good to be in that state as what we will expect to see from that point on. Given that we're a following a specific policy where we started in that state. Hm,. Does that make sense? Kay. Yeah. Very important point here, Michael, is that the reward for entering a state is not the same thing as the utility for that state. Right? And in particular. What reward gives us is immediate gratification or immediate feedback. Okay? But utility gives us

long term feedback. Does that make sense? So when reward [UNKNOWN] is the actual value that we get for being in that state. Utility [UNKNOWN] state is both the reward we get for that state. But also, all the reward that we're going to get from that point on. I see. So yeah. That seems like a really important difference. Like, if I say, here's a dollar. You know? Would you poke the president of your university in the eye? You'd be, like, okay. The immediate reward for that is one. But the long term utility of that could be actually quite low. Right. On the other hand, I say, well, why don't you go to college? And you say, but that's going to cost me $40.000. Or better yet, why don't you get a masters degree in computer science from Georgia tech, bu you can say that's going to cost me $6600. Yes, but at the end of it you will have a degree. And by the way it turns out the average starting salary for people who are getting a masters degree or undergraduate degree is about $45000. So is it considered product placement if you. Plug your own product within the product itself? No, I'm just simply stating fact Michael. This is all I'm doing. Just facts. Alright. This is called fact placement. Alright. The point is, there's a, an immediate negative reward, of say, $6,600 for, I'm going through a degree. Or maybe it's $10,000 by the time, the 15th person sees this. But anyway, it's some cost. But, presumably it's okay to go to college, or go to grad school, or whatever. Because at the end of it you are going to get something positive out of it. So it is not just that it prevents you from taking short term positive things if that is going to lead to long term negative things. I also always you to take short term negatives if it will lead to long term positives. That makes sense. What this does is this gets us back to what I mentioned earlier. Which is this notion of delayed reward. So we have this notion of reward, but utilities are really about accounting for all delayed rewards. And if you think about that, I think you can begin to see how, given you have a mathematical expression delayed rewards, you will be able to start dealing with the credit assignment problem. Cool. Okay, so let's keep going and write more equations.

U. Policies 2

So, now that we've got utility fine, and we've got this pi star to fine, we can actually do an even better job of writing out pi star. And let me do that. All right, so does this equation make sense, Michael? Let's see, so the policy, is that a star again, or is that a K. That's a star. So it's the optimal policy. All right. The optimal action to take at a state is, well, look over all the actions, and sum up overall the next states, the transition probability, so that's the probability we end up in state s prime. And now we have the utility of s prime, the problem being that that's not defined. Well, it sort of is, we defined it immediately above, at least with respect to some policy. But that's concerning because we don't know. The policy that you want to put in there is gotta be the policy that you're trying to find. Right, so in fact implicitly what I mean here is pi star. So, in fact, let me write that down that whenever you see me write from now on, the utility of a state, I'm almost always going to actually mean the utility of the state if I follow the optimal policy. We might call this the true utility of the state. I see. So I'm just going to write this off to the side here as something for you to remember. So this this says then that the optimal policy is the one that, for every state, returns the action that maximizes my expected utility. With regard to the optimal policy, it feels rather circular. It is rather circular, but you're a computationalist. You're a big fan of recursion. We just went through a whole exercise where we figured out the geometric series by effectively doing recursion. It's a similar kind of situation, for this? It kind of is. So, let me write one more equation down and then you'll be one step closer to actually seeing it. Of course, if we're in an infinite horizon with a discounted state, even though you're one step closer you won't actually be any closer. Well let's worry about that when we get there. So let me write one more equation down. We're never going to get there. It's infinitely long. [LAUGH] Yeah. Wait are you demonstrating something with this lesson by making it infinitely long? [LAUGH] I'm certainly demonstrating something with this lesson. I don't know what it is. So let me write this next equation down. So then the true utility of a state s is then, I'm just basically going to unroll the equation for utility. It's the reward that I get for being in that state, plus I'm now going to discount all of the reward that I'm going to get from that point on. Got it? All right, so once we go to our new state s prime, we're going to look at the utility of that state. Okay, that's sort of fine, modular recursion. We're going to look at overall actions, which action gives us the highest value of that. Oh I see, that's kind of like the pi star expression just above. Yup. All right, so once we figure that out, we know what action we're going to take in state s prime. We're going to discount that, because why? Because I guess that just kind of ups the gamma factor on all the rewards in the future. Right. And then we're going to add to that our immediate reward. Yes, okay I think I follow that. In some sense all I've done is I kept substituting pieces back into one another. So the true utility being in a state is the reward you get in that state, plus the discount of all the reward you're going to get at that point, which, of course, is defined as the utility you're going to get for the states that you see, but each one of those is defined similarly. And so the utility you will get for s double prime say will also be further discounted but since it's multiplied by gamma that will be gamma squared and then s triple prime will be gamma cubed, and so that's basically just unrolling this notion of utility up here. Okay so now it seems like all the pieces are in one place. Right. And so it would be nice if we were done. And I'm going to say that we're not just one step closer, but you can see an oncoming light and it is not an oncoming train, okay. So Yeah, this seems like a really important equation. It is, in fact, it's so important, it's got a name. You want to guess what the name is? Bill That's actually very close. It's Bellman Equation. Bellman equation Esquire. This equation was invented by a guy named Bellman, and it turns out to be in some sense the key equation for solving MDPs and reinforcement learning. Wow. And it's actually even more [INAUDIBLE] than it looks. But basically, this is the fundamental recursive equation that defines the true value of being in some particular state. And it accounts for everything that we care about in the MDP. The utilities themselves deal with the policy that we want to have, the gammas are discount, and all the rewards are here. The transition matrix is here, and the actions or all the actions we're going to take. So basically the whole MDP is referenced inside of here and allows us by determining utilities, to always know what's the best action to take. What's the one that's going to maximize the utility? So if we can figure out the answer to this equation, the utilities of all the states, we per force know what the optimal policy is. It becomes very easy. So we've sort of taken all that neat stuff about NDPs and stuck it in a single equation. Bellman was a very smart guy. So was he the same Bellman from the curse of dimensionality? Yes. Cool. There can be only one Bellman. [LAUGH] Actually, are there any more Bellmans? I don't think so, I think that they retired, like retiring a jersey. They retired

his name. I could've sworn that I saw one at the last hotel that I went to. It was probably the same one. Oh, I get it. Hotel, Bellman, that's really good. Very good. Okay good. Well, so now that we've killed that as much as we could, let's see if we can actually solve this equation, which since this is clearly the key equation since it has a name, okay? Yeah, that would be cool. Especially because it looks like, if you could solve this, you could solve it, right? because then you have u. You could just plug the u in and get the u out. Right. And once you have the u in, and you get the u out, then you got the policy. Right. For u. It's always been for u. [LAUGH] It's for us, Michael, it's for us.

V. Finding Policies 1

Okay Micheal, so I've erased the screen and kept Bellman's equation, the most important equation ever and we are going to solve it. Oh. So, how are we going to make that work? Okay, so how many, so we wrote this down as a utility of s, how many S's are there? s, n, m, I don't know? Pick a letter. N. N, so we N states which means this isn't really one equasion. This is how many equations? N. Yes it's N equations. How many unknowns are there? Well we have U, the Rs are known, the Ts are known, the only things that's missing is the Us. And there's, oh and there's N of those as well. Right there's N equation in N [UNKNOWN]. So we're done. Yes because we know how to solve n equations in n unknowns, right? If the equations were linear. If the equations were linear. Are these equations linear? I'm going to go with no. Why not? because the max is problematic. That's right. This max operation makes it nonlinear. So, it looks really good for a moment there. We've got n equations, n unknowns. We know how to solve that. But max makes for a very very very weird non linearity. And there's kind of no nice way to deal with this. Actually, one day, Michael, if you ask me, there is a cute little aside you can do where you can turn max into something that's differentiable. Oh. But. That doesn't actually help us here so I'm not going to go off on that aside yet. But even differentiable wouldn't quite be linear. That's right. And it wouldn't help us in this case. Yeah that's exactly right. But the fact that you can have differentiable maxes I think actually [UNKNOWN]. But also unimportant for what we're talking about now. So, we've got n equations, n unknowns, they're nonlinear, which means we can't solve it the way we want to by like inverting matrices or like. You people are in the regression would normally do but it turns out that we can in fact, do something fairly similar, something that actually allows us to solve this even though it is nonlinear. And here's the equation, or the equation, here's, here's the algorithm you use that sort of works, okay? So it's really simple. Just simply start with some arbitrary utilities. Declare those the answer and you're done. That would be one way of doing it but it turns out we can do even better. Wait. Start off with the correct utilities. That would work except we don't know what they are. Oh right. So we're going to start with some arbitrary utilities, but then we're going to update them based on their neighbors. And then we'll simply lather, rinse, repeat. Alright, so what does that mean based on neighbors? So, it means, based on the states, you're going to update the utility for a state based on all of the states that it can reach. So, let me write down an equation that tells you how to update them and then maybe it'll be clear, okay? Yep. So we're going to have a bunch of utilities since we're going to, we're going to be looping through this. And let's just say every time we loop through, that's time t. Okay? So we know what the equation for utility is where the utilities are. It's just the equation that's written up here. So it's r of s plus gamma times the mass over a of. The expected utility. Right? Except we have some estimate of the utility at time t. Okay? and, probably the right thing for me to do would be to write this like as you had or something like that. This is my estimate of the utility. And so now using this, I'm going to want to sort of update the, you know all of my utilities to make them better. And it turns out I can do that by simply doing this. So I'm going to update at every iteration update utilities based on neighbors I'm going to update at every iteration, at every iteration my estimate of the utility of some state S by simply recalculating it to be the actual reward that I get for entering state S, plus the discounted utility that I expect given the original estimates of my utility. Does that make any sense at all? Yeah, I guess so, so. So the s, okay, so, so we need the whole u hat t. Mm-hm. Like at all states, because we're not just using the values that state s to update the values that state s. We're using all the values, at the, at the previous time step to update all the values to the current time step. Yep. So, so all these n equations, they're all tangled together. Right, because of this This, expectation. So really, just to make it clear, this should be a summation over s prime.

W. Finding Policies 2

So, I'm going to update the utility of s by looking at the utilities of all the other states, including itself, as prime. And weight those based on my probability of getting there given that I took an action. And what action am I going to take? Well, I'm going to take the one that maximizes my, expected utility. So it's sort of like figuring out what to do in a state assuming that this uhat really is the right answer for the future. Right. Now why is that going to work? So I just made up the uhats, right? They started out as arbitary utilities. The next little step about updating utilities based on neighbors makes some sense because effectively is any state that you can reach. Which is determined by the transition function. But, all I'm doing is reaching states that are also made up of arbitrary utilities so, arbitrary values. Why should that help me? Well, it's because of this value right here. This is actual truth. The actual word I get for entering into a stage, is a true value. So, effectively what's going to happen is I'm going to be propagating out the true value, or true reward, the true immediate reward that I get for entering into a state, say. Through all of the states that I'm going to see and propagating that information back and forth. Until ultimately I converge. Still not obvious why we should converge, because we start off with an arbitrary function and it seems like that could be really wrong. So we're like adding truth to wrong. Yes but then, the next time around I'm going to, I've been adding truth twice to wrong, and then more truth to wrong, and more truth to wrong. And eventually, I'm going to be adding, more and more and more and more and more and more and more and more and more truth, then it will overwhelm the original wrong. And is that, does it help that the wrong is being discounted? Yes, it helps that the wrong is being discounted. Does it help that the wrong is being discounted? I actually don't know that matters. I'll have to think about that for a moment. It certainly doesn't hurt. But I, I guess the way that I think about this, I mean there's

an actual proof you can look it up but in the end I think, I, I tend to think of this as a kind of simple contraction proof that effectively at every step you have to be moving towards the answer. Because you've added in more of the reality, more of the truth. And if you remember the utility of a state is just all of the rewards that you're going to see. So, basically at every time step you have added the reward that you're going to see, and then all of the rewards you're going to see after that. And so you've gotten a better estimate of actually the sequence of rewards you're likely to see from the state. And if that gets better for any of the states, then eventually that betterness will propagate out to all the other states that they can reach or can reach them. That will keep happening and you'll keep getting closer and closer and closer for the true utility of the states until you eventually run out of closeness. Cool. Does that make sense at all? Well does it also help that the gamma is less than one. Yeah it does. the, the way I like to think of this as, is as a sort of contraction proof, that makes, if you've heard of those. So, the basic idea here is that you start out with some noise, but at every step, you're getting some truth, and that truth gets added, and then, the next dideration, more truth gets added, and more truth get, gets added. So, as you have some estimate of some particular state S, you get to update it based on truth. It's actual reward. And you bring in more truth from the other utilities, as well. As this particular utility gets better. That closeness to the true utility then gets spread to all, from, to all the states that can reach it. And because the gamma is less than one. You basically get to overwhelm the past in this case which is the original arbitrary utilities. And so as you keep iterating through this, the latest truth becomes more important than the past less truth. And so you are always getting closer and closer and closer to the truth until you eventually you do. At least that's the intuition that I like to think of. Yeah I, I kind of get that as an intuition, though I'd probably be happier going through the math, but. Well, we could do that, and by we, I mean the students can do that by actually reading the proof. All right. Okay, so cool. So, this right here is a an easy way to find the true value of states. And you do it by iterating and it kind of has a name. It kind of has a name. Yeah, what do you think the name could be? Bellman. Bellman's algorithm. No. No though that's probably reasonable. Utility iteration. Yes, except utility sounds better if you say value, so it's value iteration. And it works. Remarkably well. So, and it doesn't, doesn't give you the answer but it gets you something that is closer and closer to the answer. Right. And, eventually it will converge. You'll get so close to converging it doesn't matter. And, once you have the two utilities, if you recall. We know how to define the optimal policy in terms of utilities. So, if I give you the utilities, then the optimum policy is just, well I'm in a state. Look at all the states I might get to. Figure out the expectation that I'm going to get for a given action. Pick whichever one is maximum and I'm done. So solving for the utilities are the true value of a state. Is effectively the same thing as solving for the optimal policy. Hm. Excellent. That's cool. I think so.

X. Finding Policies 3 Question

Okay, Michael, so you seemed like you knew what you were doing so I thought I would verify that by giving you a very easy quiz. This doesn't look so easy. It is easy, so here's the quiz. To help you I've written up both the Bellmans equation and the update that we would give to utilities. I neglected to write the little hats and everything because I just don't think you need that but these are the two equations that you need to be able to think about. This is just what we've written up before. And I've written down the grid world we've been playing with the entire time. And I want you to figure out how value iteration would work from the first iteration and the second iteration for this particular state here that I marked with an X, okay? Okay. Alright, and there are a little more information you need to go. Gamma in this case is going to be equal to one half. Mainly because it's easy to do the math if you do it that way. The rewards, just to remind you, for all of the states except for the two goal or absorbing states is going to be minus 0.04. And my initial arbitrary utilities for all of the states, is going to be 0, except in the two absorbing states where I already know the utilities are 1 and minus 1 respectively, Okay? Okay. You got all that? I think so. You sure? [LAUGH] I feel confident that I'm going to slip up, but Okay, yeah, I think I can take a stab at this. Alright, so gamma's one half, rewards are minus 0.04, arbitrary starting utilities at times 0 is 0, except here at the absorbing states. Tell me how the utility here will evolve after one step, or one iteration, and two steps, or two iterations. Okay then, go.

Y. Finding Policies 3 Solution

Alright, Michael. What's the answer? I think I've already made a mistake. Okay. What's the mistake? Suggesting that we do a quiz. [LAUGH] I'm pretty sure that's always true. Unless Michael, by doing a quiz now and suffering pain, you, in the future, are a better person. In which case, you have made the right long term decision. I see. You're, this is a MDP metapoint you're making. Mm-hm. Alright. So let, but let's, let's just buckle down and do this thing. So. Okay. Alright, so at that state x, we have to consider, according to the equation, we're going to do U sub one at x. And that's going to be the reward at x, which is minus .04. Mm-hm. Plus gamma, which is a half. Yes. Feel free to write these things down. Okay, so, let's see. It's going to be minus .04 plus one half times. Alright. So now we need to do four different actions. Right. So, I don, I would make like a brace-y thing at this point. Not a bracket, but a brace. Alright? Or I could do a bracket, because you're going to notice immediately that it's obvious what the right action is. Okay, alright. Well, we know that the, the right action's going to be to go to the right. Yeah, but even then, you know you don't have to do the rest of the computation because my first guess at all the utilities is that they're 0. Which means you're always going to want to take the action that gets you to plus one with the highest chance. Right. So there's just no point in. I see. Okay. Fair enough. Thank you for. Okay. The shortcut. So, so we only have to do the one action which is to the right. Mm-hm. And so if we go to the right, there's three possible next states we could go in. Yeah. One is back to x, which has a value of zero. Mm-hm. One is to the thing underneath of x, which has a value of zero. And then the last one with probability 0.8 needs to go to the plus one. Which is. 0.8 times plus one. Which is 0.8. 0.8. Okay. And that is? Okay. So 0.8 times a half is 0.4 minus 0.04 is 0.36. Yep. And that is correct. Okay, so to do the same thing for you too, we're going to need the U1 value for a bunch of other states, seems to me. Maybe, so let's right that down. So we know for now the utility here is .36 right? Yeah. And you're saying that in order to do U two, I'm

now going to have to, you know, I was able to avoid doing some of the math before because, these are all zeroes. So it was just easy to do. But when I went right, I either stayed where I was, went to the plus one, or I ended up going down. Well I think by the same argument that allowed us to cheat our way out of to cheat our way out before. Okay, It's still going to be best to go to the right, so. Yeah but I know that but the value itself is going to depend on the value in several other states. Yeah well how many other states? Oh, just that one. Just this one, so what was U one of this state? I see. So, presumably, we want to avoid falling into the pit, so the, the best thing we can do is bash our head against the wall, Mm-hm Which will get us a -.04 for that statement. Right. Okay, alright, so maybe this isn't so, so bad. Mm-hm. So now for our u2 values we need -.04 plus a half times point, point one times point 36. Plus. Mm-hm. 0.1 times negative 0.04, so that's minus 0.004, plus 0.8 times one. Oh, which is 0.8 again, just like it was last time. Yep. And I get 0.376. Which is what I get by also getting out your calculator. Okay, so 0.376, and you can imagine how we would do that on and on. I want to point something out, Michael, which is that. You decided to figure what the true utility was for the state under X by bashing your head into the wall. But you know that based on the discussion we had earlier that actually the optimal policy would involve going up instead of bashing your head into the wall. What you did at that point was in fact right because everything else in this utilities were all zero. The best thing you could do is avoid getting, avoid ever falling into minus one, so the policy of the very first of bashing your head into the wall, is infact the right thing to do at that point. But what youll notice is that next time around the utility for the X state Is bigger than zero. In fact, will keep getting bigger and bigger than zero. As you can see, it went from .36 to 0.376. Which means that at some point it's going to be worthwhile to try to go up instead of bashing your head into a wall. I see. So this, so this is kind of cool that it works. But, it does seem like a really roundabout way of getting there. I mean, is there some way that we could some, I don't know, maybe take advantage of the fact that there's not that many policies? Yeah, so you actually said something, fairly subtle there, so let's see if we can unpack it. So, lemme point out two things, which I think will get us to what you're answering. The first is. Do you realize that the reason that the value iteration works is because eventually value propagates out from its neighbors, right? The first time we can calculate the .36 without really worrying about anything around it because the utilities are all zero, and in fact, based on the initial utilities. Even for this state here, we do the wrong thing. But after some time, this state becomes a truer representation of its utility and, in fact, gets higher and higher. The right thing to do here will go up. You'll also notice that after another time step I'm going to need to start looking at the value of this state as well. Alright. So, eventually I'm going to have to figure out the value, the utilities or the values of all of these states and this plus one is going to propagate out towards the other states. Where this minus one will propagate out less because you're going to try to avoid falling in there. So that makes sense, right? But what's propagating out, Michael? What's propagating out is the true utilities, the true values of these states. But what's a policy? A policy is a function from what to what? States to actions. Right, a policy is the mapping from state to action. It is not a mapping from states to utilities. No, that's what U is. That's what U is. Or that's what you are. So, [LAUGH], so if we have U, we can figure out pi, but U is actually much more information than we need to figure out pi. If we have a U that is not the correct utility, but say, has the ordering of the actions correct, then we're actually doing pretty well, right? It doesn't matter whether we have the wrong utilities. I see. Uh-huh. You'll remember we did this in the, the first third of the class as well when we noticed that we were computing in the Bayesian learning case actual probabilities. But we don't really care about actual probabilities. We just care that the labels are right. There's a very similar argument here. We don't care about having the correct utilities, even though by having the correct utilities, we have the right policy. All we actually care about is getting the right policy. And order matters there rather than absolute value. Does that make sense? Yeah, that's interesting. It's almost kind of like pi is more of like a classifier, right? It's mapping. Inputs to discreet classes and the user kind of like, more like regression where its mapping these states to continuous values. Right and given one, given the utilities, we can find pi, given pi there's an infinite number of utilities that are. Consistent with it. So, what you end up wanting to do is get a utility that's good enough to get you to your pie. Which is one reason why you don't have to worry about getting the absolute convergence in [UNKNOWN]. But it gives us a hint of something we might do that's a little bit better with the policies that might go faster in practise. So, I'm just going to take three seconds to give you an example of that. Okay? Awesome.

Z. Finding Policies 4

Okay so Michael let's see if we can do something that might be faster and just as good as what we've been doing with value iterations. And what we're going to do is we're going to emphasize the fact that we care about policies, not values. Now it's true given the true utilities we can find a policy, but maybe we don't have to find the true utilities. In order to find the optimal policy. So, here's a little sketch of an algorithm okay, so it's going to look a lot like value to ratio. We are going to start with some initial policy, let's call it pi sub zero and that's just going to be a guess, so it's just going to be an arbitrary setting of actions that you should take in different states. Then we're going to evaluate how good that policy is, and the way we're going to it at time T. Is to calculate its utility, which I'm going to call U sub t. Which is equal to the utility you get by following that policy. Okay? And I, I'll show you in a moment exactly how you do that, alright? But I just want to make certain that you, that you, you kind of buy that maybe we can do that. So, We have, given a policy, we ought to be able to evaluate it by figuring out what the utility of that policy is. And, again, we'll talk about that in a second. And then, after we know what the utility of that policy is, we're actually going to improve that policy in a way similar to what we did with value iteration, we're going to update our policy, Time T plus one, to be the policy. That takes the actions that maximizes the expected utility based of what we just calculated for [INAUDIBLE] of T. Now notice it will allow us to change Pi over time because image that we discover that in some state that actually there is an action that is very good in that state. That gets you some place really nice gives you a really big reward and that went on and you do fairly well. Well then all other states that can reach that state. Might end up taking a different action than they did before, because now the best action would be to move towards that state. So these two steps will actually, or can actually lead to some improvement of the policy over time. But the key thing here is we have to figure out exactly how to compute u sub t. Well, the good news is we know how

to do that, and it's actually pretty easy. And it boils down to our favorite equation, Doman's equation. [LAUGH] So our utility at time t, that is the utility that we get by following a policy at time t, is just well, the true reward that we're going to get by entering that state plus gamma times the expected utility, which now looks like this. There, does that make sense? Do you see that? Okay, hang on, it looks a little different from the other equation. So did you mean for it to have T UT is defined in terms of UT and not UT minus one? Yes. Okay, that's interesting. And the max is gone but instead of max, there's a policy. Stuck into the transition function. Yep. A choice of action is determined by the policy. Right. And that's actually the only difference between what we were doing before is that rather than having this max over actions, we already know what action we're going to take. It's determined by the policy we're currently following. Okay, but isn't this just as hard as solving? The thing with the max, you said? Well, what was the problem that we were solving before with the max? That was the Bellman equation. Yes, but we were solving a bunch of equations. How many of them? N. So we were solving n equations, and how many unknowns? N. What's the difference between this, n equations and n unknowns, and the other n equations and n unknowns? Well, n is the same. Mm hm. There's no max, though. There's no max. And what was it that made solving that hard before? It made it, the max made it non linear. The max is gone now. You're saying this is, this is a set of linear equations? Yeah. Because, well, there's, there's just a bunch of sums. And the pi is not like some weird function. This is just effectively a constant. I see. So now, I have n equations, and n unknowns. But it's in linear equations. And now that I have in linear equations and unknowns, I can actually compute this is a reasonable amount of time, by doing matrix inversions, and regression, and other magic hand wavey things. That's very slick. Yeah. It's seems, it's still more expensive than doing the valued [UNKNOWN], I guess. Yeah, but you don't have to, perhaps, do as many iterations as you were doing before. So once you've evaluated it, which we now know how to do, and you've improved it, you just keep doing that until your policy doesn't change. Very cool. Mmhm. And this does look a lot like value iteration to you, doesn't it? Yeah, though is seems like it's making bigger jumps somehow. It is, and that's because instead of making jumps. In value space, it's making jumps in policy space. Which is why we call this class of algorithms Policy Iteration. Cool. Right. Now, this inversion can still be fairly painful, it's, you know, if we don't worry about being highly efficient, you know, it's roughly n cubed, and if there are a lot of states, this can be kind of painful. But it turns out there's little tricks you can do, like do a little step evaluate iteration here for a while to get an estimate and then, you know, kind of cycle through. So there's all kinds of clever things you might want to do, but at a high level without worrying about, you know, the, the details of constants, this general process of moving through policy space. And taking advantage of the fact that by picking a specific policy you're able to turn your nonlinear equations into linear equations turns out to often to be very helpful. So, is it guaranteed to converge? Yes. Nice. Well there, that was easy. I'm, I'm not going to go into it but, you know, there's a finite number of policies. You're always getting better so eventually you have to converge. It's very similar to the, or at least intuitively it's very similar to the argument you might make for [UNKNOWN]. Cool.

AA. Wrapping up

Okay Michael, so, this is it. Why don't you help me remember what we learned in this one marathon session that was not all distributed over multiple months. [LAUGH] Sure. Or years. Mm hm. So, MDPs. So, we talked about mark-off decision processes, and we said what that meant. [LAUGH] Okay, so, that's the first thing we did is we talked about MDPs. And that, that MDP consists of states and rewards and actions and like that, transitions, discounts. So you said discounts and I just want to point out that there are some people who think that that's a part of the definition of the problem and there are some people who think that that's more of a parameter to your algorithm. okay, alright. So there's some people who think of it the right way. Like me. And some people think of it the wrong way. So, I tend to think of the discount as being something that you're allowed to fiddle with, as opposed to it being a fundamental part of the. Then why not fiddle with the rewards? Sure. You are allowed to fiddle with the rewards. Oh! You are very open minded! I am open minded, i believe that rewards should be able to marry other rewards. In any case, the important thing here is that there is some under lying process that you care about that is suppose to represent the world, states, rewards, actions, and transitions and capture that fairly well. Discount is, in some sense, an important part of the problem, because it tells you how much you want to care about the future versus the past. But it's reasonable to think of that as something that you might want to change outside of the kind of underlying physics of the world. I see. Okay, that's fair. Yeah, in some sense, the states and the actions and the transitions represent. The physical world and the rewards and the discount represent the kind of task description. Right. And of course, you say that, but if you, you could decide to define states differently, and doing that would impact both your actions and the transition function and so on and so forth. But the basic idea is right, which is, there's some underlying process we're trying to capture, and I think it's exactly right to say, states, actions, and transitions. Sort of capture that. And rewards and discounts capture more about the nature of the task, you're trying to do in that underlying world. Okay. And in, in that context we talked about two really important concepts. Policies and, value functions? Mm-hm. Which we sometimes call utilities. Right. And how do utilities differ from rewards? The utilities factor in the long term aspects, and the rewards are just telling you the moment to moment. Right. Utilities are like a group of rewards. Like a gaggle. Or a murder. Of crows. So, that we talked about how we can assign value to an infinite sequence of rewards. Mm-hm. But it helps if we use discounting to do that so that we don't get infinitely large sums. And that allowed us to deal with infinite sequences, but threat them as if their value is, in fact, finite, thus solving the immortality problem. [LAUGH] And, let's see, then we kind of. And the stationarity was a really important part of that. Yep. In fact kind of drove everything. Yeah. And all these things were tied up together in the bellman equation itself. Yes which is an awesome equation and deserves capital letters. [LAUGH] Is that it? And then, well then we solve the Bellman equation using value iteration and policy iteration. Oh, yes. I think that was it. Alright, so are any of these polynomial time algorithms? Well, the way we've been talking about them, no, but you can map these into linear programs and turn them into polynomial problems, or polynomial. So, yes these problems can be solved that way. But actually, that reminds me, of something that we haven't said and that we haven't learned today, that I think is worth mentioning. Which

is we been talking about, you know, this section of the class is about over the course is about reinforcement learning. But, we actually haven't done any reinforcement learning here because we know everything, we know the states, we know the rewards, we know the actions, we know the transitions. We have some discount, we have been solving MDP's, but that's not quite the same thing [SOUND] as doing reinforcement learning, however, it's very important to do these things, to make it easier to think about how reinforcement learning works. So, in reinforcement learning, the difference is, you don't necessarily know, the rewards and the transitions or even the states and the actions, for that matter. I see. So when are we going to get into reinforcement learning then? Next time. And when I say we next time, I mean you next time. That's your assignment. Learn all about reinforcement learning and talk about it next time. All right. I gotta go and get to work then. Okay. Well, you have a good one Michael. Thanks for all this. It's really cool. It is cool. Bye. Bye. Bye.

AB. Reinforcement Learning

Hello Charles. Hi Michael. How are you doing? I'm doing okay. Good. It's you know, exciting to be getting to talk about reinforcement learning. Mm. Reinforcement learning. It's my favorite type of learning. Is that true? It is true. Wow. I like machine learning. I like machine learning too. But of all the kinds of machine learning, reinforcement learning is my favorite kind of learning. So, let's, how bout, how bout, giving the opportunity for the students in the class to learn about reinforcement learning by having us tell them about it. Oh, let's do that. You first. Alright. We're going to build up on the stuff you told us about last time. You mean the fantastic, well defined, and well formalized stuff that we talked about last time? Yes, it was fantastic and it laid the groundwork for what I would like to talk about. So you set up Markov decision processes, and I'm going to talk about what it means to learn in that context. Excellent. I find it useful to start off by thinking about a reinforcement learning API, like application programmer interface. So what you talked about is, is this box here. The idea of being able to take a model of an MDP, which consists of a transition function, T, and a reward function R. And it goes through some code and, you know, it comes out and a policy comes out. Right? And a policy is, is like pi. It maps states to actions. And that whole activity, what would you call that? What would you call the, the, the problem of, or the approach of taking models and turning them into policies? Maybe I'd call them planning. Yeah that's what I, that's, that's what I was hoping you would say. Mmhm. Alright, now we're going to talk about a different set up here. We're still interested in spitting out policies, right? Figuring out how to behave to maximize reward. But a learner's going to do something different. Instead of taking a model as input, it's going to take transitions. It's going to take samples of, being in some state, taking some action, observing a reward and observing the state that is at the other end of that transition. Alright? And we'll, I put a little star on that to say well we're going to see a bunch of these transitions. Mm. And using that information we're going to instead of computing a policy, we're going to learn a policy. I see. So we call that learning? Yeah, or even reinforcement learning. Mm. By the way, what makes it reinforcement learning? That's a question. [LAUGH]. It's not a good question, but it's a question. I was, I was going to say good question, but I'll, well maybe, maybe there, it's not that it was a bad question, it's that I don't have a particularly good answer for it. So, maybe we need another slide to discuss that. Okay.

AC. Rat Dinosaurs

All right, so let's do a brief aside about the [LAUGH], this is like a bad history of reinforcement learning so it's not that it's a bad history, it's just badly told. So but the basic idea is this, once upon a time and we're going to call it, say the 1930s people observed. That if you put an animal, this is supposed to be a rat, in, in a, you know, box, say. And give it choices of looking in place B and place A and, say, one of them has cheese in it and the other one doesn't, but it can't see which because, you know, the doors are closed. And and let's say that we, we consistently do something like this. We turn on a red light whenever the cheese is in the A place. And we turn on a blue light whenever the cheese is in the, B place. And what you observe is if you do this consistently, then what you find is that if the animal sees the stimulus like the red light going on and it takes an action like peeking inside the, the or sticking it's head inside the box A and it gets a reward which in this case would be getting to eat some cheese. That that set up strengthens its response to the stimulus. In other words, in the future when it sees the red light it's going to be more likely to go in and look in box A. And this notion of strengthening, you can think of it as reinforcing the connection. Reinforcing just means strengthening. Hm. So, this was studied for a long time and there's, there's tremendous amount of interesting research about this, but if you start to, you know, think about it as a computer scientist. A natural way of, of kind of thinking about what this problem is, is that you know, a stimulus is kind of like a state, and an action is kind of like an action, and a reward, well, I kind of stole all these words but it's kind of like a reward. And it leads you to this idea that what you really want to do. Is figure out how to choose actions to maximize reward as a function of the state. And so we started to take this concept and we called it reinforcement learning because that's what the psychologists were calling it. But, for us it just means reward maximization. This notion of strengthening is really not part of the story for us at all. So, we're really taking this word and using it, you know, wrong. Hm. Well, that all makes sense except for one thing, which is the idea that this happened in the 1930s. Because we all know that rat dinosaurs did not exist in the 1930s. Yeah, they, well they went extinct in the 40s. Hm, hm. There's this one little epilogue to the story which I find really kind of amusing, and that is the computer scientists did pretty well at figuring out algorithms for solving problems like this. And the psychologists really do care about how stimuli and, and, motor actions and reward all Interact with each other and they've turned to the computer scientists to say, how might the brain be doing this? What, what is the problem that the brain actually might be solving? And so, guess what word [LAUGH] they borrowed back when they started to think about these things. Reinforcement. So, now they talk about reinforcement learning and they don't mean reinforcement, either. They mean reward maximization. So we won. [LAUGH] Yeah, I do not think that was really our plan but but it's, it's nice to know that we had some you know, impact. Well, that's very good you brought planning back into it and so they learned because of our plan, that's pretty good Michael. [LAUGH] All right, so that's the end of this aside. Okay. Let's go, let's go, let's go back to learning things.

AD. API

All right, so now that we're back from our aside, let's go back to thinking about this applications program interface. So I talked about planning and I talked about learning. And it turns out that there's two other sub-processes or sub-components that might be useful to think about that kind of relate these quantities together differently. One is the notion of a modeler. What a modeler can do is take transitions and build a modeler out of it. That makes sense. And a simulator is kind of the inverse, where you can take a model and you can use it to generate transitions. You can actually kind of imagine running around in the world, Just by simulating that row. This is generally not a very hard thing to do. Though there's certainly applications of reinforcement learning where this simulator is extremely expensive. Because it's simulating a lot of things in the world. And this modeling problem you can think of again as a kind of machine learning problem. Right? Trying to map this kind of information into models. So you call them sub-processes. Does this mean that one way to do learning would be to do modeling and simulating so that I had models. And I knew what the reinforcement function was, and then I could just do planning? Yeah, that's a really good idea. In fact, let's look at different ways of gluing these things together. Okay.

AE. API Quiz Question

So your suggestion was to take a modeler and use it to take turn transitions into a model, but once we have a model, you already told us last time how we can use a planner to spit out a policy. You didn't talk about those planners. What are, what's the name of the algorithms that you described last time? Value iteration and policy iteration. Yes, right. So, so what you can do is, is run either of those algorithms in here. They both have the same API right, they both take models and spit out policies. True. Alright, so, so let's do this as a quiz. We'll say, let's use your idea of mapping transitions to model to policy. And what would you call this whole box? Right? And so as a whole box, it takes transitions in and, and produces policy out. So it is solving the reinforcement learning problem. But it's taking a very particular approach to it. But let's contrast it with the, with kind of the opposite idea. Which is, we can also map a model through a simulator into transitions. And then if we have the ability to do reinforcement learning. We can use, turn those transitions into a policy. So again, as a, as a, composed system. This is turning a model into a policy, so it is a kind of planner, but it's a planner that uses a learner inside and this is a learner that uses a planner inside. So just, as, just out of curiosity I would like to, just ask people what they'd want to call these. I'm not going to grade these, but I just I'm just interested. Like, what would you call this approach? Does that make sense? Just type it in the box. Let's pretend it makes sense. Alright. Go.

AF. API Quiz Solution

All right Charles, so what would you, what would you call let's, let's let's think about this bottom one first. Okay. So what would you call an approach to reinforcement learning that what it does is it builds a model and then plans with it? A planner? Well, it's not a planner. I mean, the planner's inside of it. Sure. The overall system, this sort of blue box. Turns transitions into policies so it's kind of a reinforcement learner. Yeah. But it's one that builds a model inside. I would call that. You know what I would call that, I would call that model-based learning or model-based planning. Actually it's called model-based reinforcement learning. Mm-hm. So this is, this is, in fact, my, you said, reinforcement learning is your favorite kind of learning? Mm-hm. Model-based reinforcement learning is my favorite kind of of reinforcement learning. Mm. But we're not going to get to talk about it very much, so I wanted to at least have this slide to give people a chance to, at least, you know, these are all pieces that you can imagine doing. Right? This, this piece you can think of as being what we did in, when we're talking about supervised learning. And this piece is what you talked about last time. So you know, you could build a model based reinforcement learner that way. Yeah, that makes sense. Alright, how about this other idea, where, where you start off with a model and then you pretend you don't have a model, you pretend you're just in a learning situation by turning that model into, into transitions just by simulating them. Then the learner interacts with the simulator. And then spits out a policy. Well, I could come up with one of two answers. Okay. So I could try to do pattern matching on the answer to the second one. And since that one's model based, then this one's transition based, which is kind of cute. Or I could say, well one difference between at least inside the kind of blue box is it's sort of a model free reinforcement module. because the learner does not ever get to see the model. This is true, the learning piece is model free but we're using model free learner in service of planning. Okay. So, I don't know, I don't have a good this like model free planner, or an RL-based planner, maybe. Like I said, I wasn't going to grade it, I just was kind of interested to see what, what you'd say. Hm. Well I, I was pretty, I felt pretty good about the model-based RL one. Yeah, well this is the actual term that's used in the field. I'm, I'm sure there are some terms that are used here but nothing, nothing really very consistently. Hm. What if I called it the blue box planner? You could call it that, and, but had I, had I drawn it with a black box, it would, it had a better name. mm, oo, that would have been really cool, black box planner. I like that. Okay. But but I want to point out that one of the most successful applications of reinforcement learning are least most celebrated was Jerry Tassara's backgammon playing program, which used exactly this approach. Backgammon is a board game. So we have a complete model of it, but we don't really know how to plan it, it's a very complex, very large state space. So what he did is, he actually used a simulator backgammon, to generate transitions, and then used reinforcement learning approach TD, to, to create a policy for that. So it, his TD Gammon system followed exactly this, this overall pattern. Yeah, it was very influential work and generated many mildly embarrassing Master's theses. [LAUGH] I can only think of one. Oh, actually I can think, that's not true, I can think of two. Oh really? Well, which one are you, you're thinking of yours. Yes, I'm thinking of mine. I try not to think about it. The, the other one is, Justin Boyin, who's a good friend. And, a highly influential reinforcement learning contributor, and now Googler. Who, his Masters thesis was also, it's basically another TD. He did backgammon with RL. Oh, I didn't realize that. I, I actually like him. I think he's very cool. [LAUGH]. So, I'm sure that, his was not mildly embarrassing. I mean, you know, everybody's master's thesis are mildly embarrassing because you're, you're struggling to learn about how to talk about research, and so you're not going

to get it perfect. Yeah, and in my case, it was it definitely model free. Right, the only non-embarrassing master's thesis that I'm aware of. Shanon's. Oh sure, alright, well Shannon. What was his master's thesis? It was information theory. Oh, yeah. It's pretty impressive. Yeah. Rob Schapire's master's thesis is pretty cool. I have no doubt. The boosting guy. He did, he did pom de pe learning. For his master's thesis? Yeah. With the diversity representation. I am a terrible human being. [LAUGH]. I mean wow. I mean it is true though, you look at someone like Shannon and you just think to yourself, oh, for his Master's thesis he did, he invented information theory. [LAUGH] You know, like wow, I wonder what he did for his PhD thesis? You know what he did for his PhD thesis? Nobody knows, nobody cares, because he based it on. [LAUGH] He, he had already done the information theories. They're like here, fill out a piece of paper, you can pick up a PHD. He did something on AI, I can't remember what it was, but it was, it was totally unimportant and nobody cares about it. But information theory? Yeah. [LAUGH]. Thanks. All right, well, yeah, okay. [LAUGH]. II think, I think Schapire's PHD was boosting. Thanks Rob. Make us all look bad, why don't you.

AG. Three Approaches to RL

Alright, we're going to drill down and talk about a specific reinforcement learning algorithm in just a moment. But I wanted to remind everybody about, some of the quantities that Charles introduced in the lesson last time when he was talking about MVP's, and use them to describe three different approaches for solving reinforcement learning problems. So this first box here pi, maps states to actions, and what did you call this Charles? A policy. That's right. And reinforcement learning algorithms that work directly on trying to find the policy, are called policy search algorithms. So the good thing about policy search is, you're learning the quantity that you directly need to use. Right? You're learning the policy. That's supposed to be the output. So that seems like a really good thing. Unfortunately, learning this, this function is very indirect. We don't get direct access to, I was in this state, what actions should I have chosen? Right? So this is, what did you call this, Charles, the the temporal credit assignment problem, right? Right. So, the data doesn't tell you what action to actually choose. Right. And this is why it's not exactly like the way we've formulated supervised learning in the past. Well, at least if we're trying to do policy certs, that's right. But what we're going to do is now consider, well, maybe that's not the quantity that we want to learn. Let's, let's think about learning this function U, which you had said maps states to values. So, what was this guy? U is a utility. Yeah. Yeah the true utility of the state, sometimes I, I think I called it the, the value of the state. Yeah so, so, so sometimes it's referred to as a value function, and learning methods, reinforcement learning methods that target that as, as what they're trying to learn directly, are called value function based approaches. Mm-hm. And, let's say that we, that we try to learn this, we're trying to learn to map states to values, well the good news is, at least if we're acting in the world, we're getting to see, okay I was in some state, I took some action, duh, duh, duh, and I can, I can observe the values that actually result from that, and maybe use that to, to make updates. So you can kind of imagine learning this, so that the, the learning's not quite as indirect. How do we use this to decide how to behave? So we, we need to turn it into a policy. Mm hm. And we didn't quite, we sort of talked about how to do that, in terms of, of looking at the Bellman Equations. We're going to, we're going to, it turns out it's actually kind of hard to use U directly to do that. But we're going to talk about a different form of the value function that's going to make that easy. So this is actually, it turns out it'll be a relatively easy kind of argmax operation, once we have the right kind of value function. Okay. So, there's some computation we have to do, and there's some indirectness to the learning, but, you know, it's okay. Alright, so the other quantities that you told us about were T, which is, I think of it as the transition function, I think you had a different name for it. I just call it the transition model. The, the model, right, yeah. So this is the model, and and the, the R is the reinforcement function, the reward function. Mm-hm. And what do these things do, they take states and actions and the transition function, or the transition model, predicts next states, maybe probabilistically, and the reward function tells you, next rewards, but just rewards. And so this is a model, jointly, and I already mentioned this, but we call methods that learn this quantity, learn these, these functions and then use them to do decisions, model based reinforcement learners, so how do we go from T and R to something like U? Well if we had those, if we had T and we had R, and we could see the S's, and we, and the actions we took, then we could do you know, something like value iteration we did before the learned values. Right, which value iteration was used to solve the Bellman Equations. Right. So that is somewhat heavy weight computation to do, but it's doable. So, what would happens over here, is we actually have fairly direct learning. Why is that? Because when we're trying to learn the transition and reward function, we get state action pairs as input. And then when we receive next state reward pairs as output, so we can solve this as a supervised learning problem. Hm. So, so learning is rather direct, the usage of this is a little bit computationally complex because you actually have to do the planning and then the optimizing to actually develop what you're, you know the policy doesn't come directly out of that. But but this kind of gives you a sense of three different ways, three different ways, places that you can target the pieces of the MVP so that you can do reinforcement learning. I like that. Now, we're going to focus on this, this middle piece. Partly because, you know, it's kind of the Goldilocks situation. It's you know, the learning's not so indirect and the usage is not so indirect. There's just been a tremendous amount of work focused on, on value function based approaches. And it's, you know, remarkably simple it turns out, if you do it right. And also very powerful. That there's lots of ways to use these simple ideas to, to learn hard problems. So, I think this is a good place to focus. Okay. I buy that.

AH. A New Kind of Value Function

So let's talk about a new kind of value function, that's going to actually make that optimization part easier and the learning part easier. So but it's really closely related to the stuff you talked about, Charles, so let's let's start with what you told us. Which is here's a definition of, of you. This is, we're going to define you. Mm. And put you in a box. It's been tried before. It can't be done. All right, so well, you know, if you, if you're a good dog, I'll give you some pie next. So U, [LAUGH] [LAUGH] U is defined for each state, the utility being the state, this is, the long-term value of being in a state is the reward

for arriving in that state, plus the discounted reward of the future, and what's going to happen in the future? Well, in the future, to leave this state we're going to choose some action, then we're going to take an expectation over all possible next states. And we're going to arrive in some next state, S prime, and then U is representing the utility of that as well. So this is, is recursive and nonlinear, but we know how to solve this, we can use things like value iteration to do that. Agreed? Agreed. Alright, and you also said, here's how you can use this quantity to decide what to do. That the policy in a state S is, well, let's consider all of the actions we can take to leave that state. We'll look to what their expected values are, so we'll iterate over all the possible next states weighted by their probability of the utility of landing in the state that we'd end up in. Mm-hm. And that, that tells us how to behave. Yes. Alright, so these are, these are the value functions, well the value function of the policy that we talked about before. Here's our new kind of value function. It is sometimes called the Q function. Though, you know, some people in the know don't like that. It's, it's called the Q function because it's the letter Q. But it's, some people have said that it stands for quality but it's just, it's just a letter in the latter half of the alphabet, you know, V was taken, U was taken, you know, W is used for weights, like, Q was available. So, it was brought to bear. So, so this is, this is a, a new definition and you can see it's got elements of the other two. Let me maybe write down what it, it, a way to interpret this. Okay. So, again, here's the, the Q function. And what we're going to think of this as, is, this is the value for arriving in some state, S. And this is, you know, this is the reward we get for, for that arrival. Then what we're going to do is we're going to leave S via action A. So we're going to add to that the discounted expected value that we get for taking action A. It's going to now drop us to some next state S prime, according to this probability, and once we land in S prime, we're going to take whichever action has the highest Q value from there, okay? So that turned out to be the value for arriving in S, leaving via A, and proceeding optimally thereafter. Like once, once we get to a new state, we're going to be choosing the best actions each time. Does that work for you? It makes sense right, because you could, U is basically defined the same way. It's the value for arriving in S and then proceeding optimally thereafter. That's right. And all the, the only thing that we're doing here is we're, we're basically tying the algorithm's hands briefly. We're saying, we're going to force you to leave via ac, via action A. After that, do the normal thing. But just for a moment I'd like you to just take action A. Mm-hm. Okay, that makes sense. So you just, you, you inserted basically a, a kind of, kind of utility step in there that I assume you're about to use in some clever way. Indeed, yes, this is going to turn out to be really helpful, because it's going to allow us to compare the values of different actions without having to actually stare at the model to do it.

AI. Value Function Quiz Question

Alright, so I just introduced this Q function. It turns out that baked into this one little Q function is everything we need for, for dealing with U and pi without knowing the transition function T or the reward function R. And to you know, to demonstrate this for you, I'm going to let you demonstrate it to me. [LAUGH] So, [LAUGH] here's what I'd like you to do. I'd like you to rewrite these equations, this U and the pi equation, in the little boxes, using Q instead of any references to T and R, alright? So, imagine that we have Q, that somebody's already solved this out for us. How can we define pi and U using Q? Does that seem okay? Seems okay to me. I think I might even know the answer. Alright, that would be awesome. So let's, let's give people a chance to think about it and then just tell me what you think. Okay.

AJ. Value Function Quiz Solution

Alright, Charles [LAUGH], let's, let's get back to the task at hand, and you thought maybe you had some ideas. So, tell, tell me, how can we define U and pi in terms of Q? Okay. Well, so I guess the first thing to observe is that U is a value. Right? U are a value. Okay. Yes. U is a value. That's correct. [LAUGH] U returns a number. A scalar, in particular, where pi returns an action. Yeah, U of s returns a scalar, that's right, and pi of s returns an action, very good. Right, okay. And, you know, we have the definition for U up there near the top, and it's just the reward and then sort of behaving optimally thereafter, where Q is a reward and then taking a specific action, and and behaving optimally there after. So, we can sort of turn Q into U, if we always pick the best action. And we know the best action is, it's the one that, you know maximizes the value that you're going to get from that point on. So, I would say that U of s could be defined as Max over a of Q, s comma a. Simplicity itself. Mm-hm. Nicely done. I have no idea how we're going to grade that automatically, because it's hard to type a's under other a's. But, yeah, that's exactly right, that we have the maximization over all possible actions of the Q value in that state. Like, that's just great! Mm-hm How about the policy? So the policy will look exactly the same, the, the, the policy that you want to follow anyway is the one that... Maximizes your value going forward except the differences. It's returning an A and not an actual value, so it should be argmax. Oh, So it's not exactly the same. Right. It's almost exactly the same. Rr… max Right, so this is, it's just so trivial, so, so Q gives us everything we need. If only we had a good way of finding Q. Yeah, if only, if only we could find a Q, a Q for you. [LAUGH] and that's going to be the essense of Q LEARNING. Oh, well done.

AK. Q Learning Question

Alright, I apologize in advance but we're going to do a quiz. Q-learning, The Quiz, you know, because quiz starts with Q. So which of these actually describes Q-learning? Is it the problem of figuring out the best line to wait in? Is it discovering when to come in for your line, when you're in a play? Is it practicing the best bank shot? Or is it evaluating the Bellman equations from data? Alright, do you want [LAUGH] are you willing to do this quiz Charles? I am, I am willing to do your quiz your quiz learning quiz. All right. All right. Let's do it then. Okay. Go!

AL. Q Learning Solution

All right, take me through. Okay. So, the first thing I want to say is, I think it's awesome that you made me do this quiz. The second thing I want to say is, in some sense, every single one of them is correct. Yay! But, you put radio buttons down, not check box buttons, thingys, so I have to pick one. So let's just go through. Figure out the best line to wait in. Well, that is a queue, queue which would make a lot of sense if I were, say, English. Yes, that's right. In fact, you are, in this picture. Oh yeah, great. [LAUGH] That's the English version of Charles, the one with the top hat. Okay. Yeah, and a cane, too. Yeah, I like it. And a cane, yes, well that is typically what I do whenever I go to London. Discovering when to come in for your line, that is also a queue-learning though a different kind of queue. que-learning. The third practicing your best bank shot, although I probably would've said break shot. Ooh, nice. That is also cue-learning. And it's the same spelling. That's the same spelling. That's unfortunate. Well, you pronounce one cue, and you pronounce the other one cue. Yes, well clearly. But, and I spelled them that way too. Yeah, and I know. I know. I can see where the emphasis is. And the fourth one is evaluating the Bellman equations from data. That is you take in the states, the actions, rewards and the next states, and you try to learn an actual Q function, which is just the letter Q. So that is Q-learning, and is the only one that is spelled correctly, and so that is the choice that I would make. Alright, so, just to be clear here. So this all [LAUGH] the purpose here other than to. Demonstrate your ability to show puns? [LAUGH] Is that is just to give the definition. So that, so what Q-learning is going to be doing is it's going to use transitions, that is to say data, to directly produce the solution to those Q equations.

AM. Estimating Q From Transitions

Alright, so what we're going to do to figure out how Q learning works, is we're going to think about what it means to estimate this Q function from transitions. So, let's just remember this is the form of the Q equation, that we've been talking about. And, we can't do this. We can't solve this, because we don't have access to R and T. All we have access to are transitions. So this a really, I guess, I'm going to guess you said this before, but when you, when you write out this equation it really jumps out at me. This is the difference between what I was talking about, solving MDPs and reinforcement learning. In solving MDPs we had R and we had T and now we do not have them, so we have to come up with some other way to solve these kinds of equations. That's right. Okay. So if we did have R and T, then we could solve this? Yeah, this is, I mean, the same things that you talked about, value iteration, policy iteration? It can be formulated in terms of Q. So it's, yeah, there's, this is easy to do, well, [LAUGH] it's polynomial to do if you have access to T and R. But but again, in the learning scenario we don't have the model. What we have are transitions. Okay, okay. So, here's how we're going to use transitions. This is what a transition is. We, we observe that we were in some state S of the MDP. And then action A was chosen somehow. And, then a transition happens. We land in a state. We get the reward for landing in that state. And we find out what state we're in. So that's, that's the transition. And what are we going to do with it? Well, what we're going to do. Is imagine that we've got an estimate of the Q function, Q hat, and we're going to update it as follows. Here's how we're going to use all these quantities from the transition. We're going to take the, the state and action that we just experienced. And we're going to change it; we going to update it; we're going to move a little bit. Alpha, this is alpha, this is called a learning rate. We're going to move a little bit in the direction of the immediate reward plus the discounted estimated value of the next state. So, we're going to take our estimate Q hat, we going to take the state that we end up in as prime. We're going to look at all the different actions we could take from there and take the maximum. So this together is kind of an estimate of the utility, right? Mm-hm. And this is the utility of the state that we're going to. This altogether is the utility of the state that, that we're in. To the state S. So this is kind of the utility of the state that we're landing in as prime. And this all together is, is related to the utility of the state, right? You can see that it's related. In that we've got the immediate reward, which kind of matches to this. We've got the discounting. We don't have the sum over transitions but we do have the max A and the lookup in the next, in the Q function. Alright so this, this is the Q learning equation. Alright, let me just say a little bit more about this, this alpha arrow notation, which I really like but is not all that standard. So if you, when I write you know something like V gets with an alpha X. What we mean is we're moving alpha of the way from the current value V towards X, which can be written this way. That V gets 1 minus alpha of V plus alpha of x. And so in particular, if you think about this as so if alpha is 0. That's sort of a learning rate of 0, which shouldn't learn at all, and in fact, if you set alpha to 0 here, it's going to zero out X, and it's going to only assign V to V, so nothing's going to change. So learning rate of 0 corresponds to no learning. And if we set alpha to 1, that's like full learning, so we forget everything that we knew and we just jumped to the new value. And that's what happens here, that 1 minus alpha is 0, so the V goes away, and we just get X assigned to V. So does that make sense? That does make sense. And and if alpha were in between 0 and 1 like one half, you're basically making V the average of the old value of V, and the new value X that you see. Good.

AN. Learning Incrementally Question

Alright, let's do a little quiz. Okay. To help us to kind of, make sense of the equations in the Q learning equation by looking at a simpler case. So let's imagine that we got some variable V and some sequencing value X and a sequence of learning rates, alpha sub t. So we're going to draw a series of these X values and use them to update the V values and the learning rate is changing over time and the learning rate is going to have, first of all X is going to be drawn X sub T. Is going to be drawn from some, the distribution of some random variable, big X. Mm-hm. And the learning rates are going to satisfy two properties. One is that the sum of the learning rates, summed up to infinity. But, the sum of the squares of the learning rate, as we go to infinity, sums up to something less than infinity. So can, can you think of a learning rate sequence that has that property? So, the one that I remember is alpha sub t equals one over t. Good. So in fact these's a whole range of possible powers of t that work here, but one over t is a good one. Why is this, why does it satisfy the properties? Well, if you sum up the values up to sum value t, sum up the one over i values, it actually acts like the natural logarithm. And so, as t goes to

infinity, the sum goes to infinity. But logarithmically. Which is still you know, infinity. Yeah. But if you look at the sums of the squares, that's actually a well known problem called the I think, the Basil problem, the Basil problem? And it turns out to actually be pie squared over six which is kind of crazy. But there it is which is a finite value. That's intuitively obvious even to the most casual observer. No I really, this, it's, it's, it's whacky. It was an open problem for a long time and then it was finally solved. And it's like, yeah sure pi squared over six. Like where's the pi in here? There's no pi in here. Mm, I didn't find pi anywhere. [LAUGH] That's neither here nor there. The whole, the point is there's, there's a bunch of sequences of learning rates that satisfy this property, and we're going to imagine that we have a sequence of learning rates that satisfies this property. We're going to be updating V sub-t, with a series of X values drawn from, from distribution big X, and what I'm asking is, what do you think this converges to? Do you think it converges to the expected value of X? Do you think maybe it just doesn't converge at all? Do you think it converges to the variance of X? Of, of the random variable X? Or does it converge to infinity, it just keeps growing without bound? All right. Does that seem like a well formed question? sure. All right. So let's give you a chance to think about it. Okay.

AO. Learning Incrementally Solution

Alright, what do you think, Charles? Okay, so I think a couple of things. Let me just talk this out loud. So we're trying to learn incrementally. We're trying to figure out, how if we just kind of keep adding these random variables in a sort of way to what you described before what we would end up at. So why do we want to choose alpha t with a particular kind of property well if you look at it, it's clear that alpha t at each time step is moving closer and closer to zero. So it means you start out learning a lot and you sort of believe things less and less and less or you let things change you less and less and less over time. So what is that likely to end up at? Well the first thing I notice that well if it's going to do that then it has to converge. Some point. It just makes sense. So I, that helps me to eliminate the second answer. But actually, that also let's me eliminate the fourth answer because in some ways they, that says the same thing. So that leaves us with the expected value of x and the expected value of x squared. For the expected value of x squared. Whichever one it is that is variance. Okay so, I'm going to go with, the expected value of x. So, the expected value of x, right. So, so it turns out that this is actually a wave computing the average by just, you know, repeatedly sampling and updating your values. And, and the way to think about it, is that sometimes the x values that we draw are going to be a bit above the average, and it's going to pull the v value up. Sometimes it's a little bit below the average, and it's going to pull the value down. But in the limit, all those pulls and pushes are going to cancel out, and it's going to settle in on the actual average value or the expected value of this random variable. Right, because in principle, you're going to see an infinite number of those things. And effectively all you are doing is adding them all up, and sort of, you know. Well you're effectively averaging them one little bit at a time. Yeah, that's a good thing about it, that in fact it's adding them up and it's computing a weighted average, but the weights are these alpha's and the weights are decaying over time so we're going to put more weight on the more recent ones, but some more weight on the further away ones but it sort of doesn't matter because the order, since we're drawing this iid, the order doesn't matter and the thing that has to converge to is the mean. Right. I like it. So let's relate that back to what we do in Q-learning.

AP. Estimating Q From Transitions Two

Alright, so this is now the Q in the equation, again. Which again is one of these alphabased things. And, just to be clear, I really do mean, you know, alpha sub t. That we're, that were doing this over time. We're updating our learning rates as we go. It's just, sometimes it's a little irritating to put that, that there. But but yeah. So let's imagine that we're doing that. We're, we're, we're using that same kind of weighted process. Bumping the values around. So, what would, based on what we just talked about, Charles, what would this actually be computing? Well, it would be computing the average value that you would get for following, you know, kind of optimal policy after you take this particular action. Yeah, that it's, it's trying to go to this expected value and, and what is that expected value. So the linearity of expectation says that we can actually move the expectation to break up, break up the sum using the expectation. So this, the expected value of the reward is actually r of s. Mm-hm. The gamma's going to come out because of linearity of expectation. ANd then what we're left with is the expectation over all next dates of the maximum estimated value of the estimated Q value of the next state. But what is this distribution over S prime? It's the distribution that is determeined by the transtiion function. SO it's this. Which is you know this. So that's good. It's I'm kind of cheating though. Do you see how I'm cheating. I think I know how you are cheating but tell me. Well so when I told the story about this alpa arrow updating thing. I said that it convergence to the expected value of this quantity here. But this quantity here... Since it's "Q hat", it's actually changing over time. Right. So this target is actually a moving target. It's changing over time. So we can't quite do this analysis because the first step is a little bit questionable. But it turns out that there really is a theorem that this simple update rule, this Q-learning update rule, this tiny little one line of code, actually solves Markov decision processes. Yeah.

AQ. Q Learning Convergence

So this is a remarkable fact about this Q-learning rule, and that is if we start Q hat off pretty much anywhere, and then we update it according to the rule that we talked about. Q for, for when we see a transition s,a, r, s prime, then we update (s,a), the Q value for (s,a), move it alpha of the way towards r plus gamma, max a of the, well basically the Q value of the state S prime. Then as long as we do that, then this estimate, this q hat S A goes to Q S A. The actual solution to the Bellman equation. And I write this with an exclamation mark, because it's like, it's one line of code! It's one line of code, like, how could you not just go out and write this right now? Hm. But the, the, the, let me just, to finish is, this is only true if we actually visit SA infinitely often. So you know, that's an important caveat. That for this to, to actually hold true, for you to

really converge to the, the solution, it has to run for a long time. It has to visit all state action pairs. The learning rates have to be updated the way that we talked about before. The next states need to be drawn from the actual transition probabilities but that's, that's cool, if we actually are learning in some actual environment and the rewards need to be drawn from the rewards function. So, this isn't so problematic. This is a little bit problematic, but it is still very reassuring, this idea that we have the right form of an update rule, so that the thing that we converge to is the actual optimal solution to the MDP. Cool. And we just have to wait til the heat death of the universe, or infinity, and then we're done. Yeah.

AR. Choosing Actions

So, Charles, I kind of cheated. Oh, tell me more. So, Q-learning isn't really an algorithm. Q-leaning is actually a family of algorithms. There's lots of different reinforcement learning algorithms, specific reinforcement learning algorithms that can be reasonably called Q-learning, and they vary typically along these three dimensions. How do we initialize our estimate, Q hat, how do we decay our learning rates, alpha sub-t? And how do we choose actions during learning? Hm. And different ways of making these choices actually lead to algorithms with fairly different behaviors. In particular when we use this in the context of an MDP, well let's, let me, let me ask you. So like, what do you think might matter about let's start with the last one, choosing actions? Well. It see, well there's a bunch of dumb things you could do, right? You could just, just pick an amption, action, action every single time, like the same action every single time independent of what you learned, that's kind of dumb. But, it seems like the obvious smart thing to do is say look, I'm learning, I'm getting better and better, so what I'm going to do is at this next time steps is I actually have to take an action. I'll just pick the action that my Q hat tells me is the best action to take. And I'm done. So all right, let me, let me see if I can capture some of what you just said there. So, one way to choose actions really badly is say, pick some action, call it a sub 0. And no matter what state your in. No matter what's happen so far, always choose that action. Mm-hm. Right, so this possibly can't work. It's going to violate the Q-learning convergence that says we have to visit each state action pair infinitely often, and update them to converge and you know and, and it makes perfect sense. Like if we never try something, like how do you know that you don't like something if you've never even tried it. Like spinach. Exactly. Another idea would be to choose randomly. And this seems kind of good and that we are going to visit. You know, all the states that are visitable and we will try all the actions that are actionable. And we could actually learn Q this way. But, as you pointed out, this is not a great idea because we may have learned Q, but we haven't really used it. We haven't really chosen actions using what we've learned, so it's like we're wise but we are impotent. No, I think it's more like we're wise but we're stupid. I mean. We're wise but we still. We know a lot but refuse to actually do anything about it. Right. In particular in some sense the only difference between the two is the the theorem, right? If I, if I'm just choosing randomly. Wha, what's the problem with them always choosing a sub 0. Well, you're don't going to converge but the real problem is that you don't learn anything. Or you don't take advantage of anything you learned. Choosing randomly is basically the same thing, you basically, you never take advantage of anything that you learn. What's the point in learning a Q function if you are always going to behave randomly, you've learned enough or you've learned but you [CROSSTALK]. [CROSSTALK] right you've actually learned the ultra policy but you're not following it so you're not actually using what you know, so you can't you know it doesn't, it doesn't work all right. So then you had another idea. Which was to use our estimate to choose actions. Yeah. And that seems like a good idea in that we will use it. Is it possible that it won't learn? Well, it will learn something. Well, yeah. It might not learn anything all that good though. So for example, what if we do something like this. So we initialize, now we're back up to this, this first point here. We initialize the, the estimate Q hat so that for every state, a0 looks awesome and all the other actions look terrible. Wait [INAUDIBLE] is that metric awesome? Or English awesome? Oh you're right I'm sorry, I didn't put units on that. That's in Chilean dollars. [LAUGH] Oh okay, so it's pretty awesome then. Okay. So, if you do that, then well let's see what happens. It's almost like always taking a0. The only thing that would save you from taking a0 forever, is if, as you take a0, you learn that, you update your Qs and you keep getting really, really bad results. Really, really bad results, in fact, worse than terrible. yes. Well let's imagine the terrible is worse than terrible. Oh. So, but you're right, yeah you're right. So, so there's, there's at least the case that if this, if this terrible value is actually lower than the value of always choosing a0, then we'll continue to use a0 forever. This is, this is called the greedy action selection strategy. Mm-mh. And that, this is the problem that runs into, it's a kind of local min. Oh, I see. Oh and oh, okay, I see that. So, you, you didn't even have to come up with this ridiculous example. If you, if you well, not ridiculous, but you, extremely. I was going to say, I'm, I'm sorry, ridiculous how? Well, I'm sorry. It, it's a ridiculous situation to be in. It's sort of the extremely unluicky example. I think what you're saying is that you don't like people from Chile. [LAUGH] Oh no I love people from, I love Chile. Especially with you know, just some good beans and some nice meat. But the thing is that you, if you randomly set your Q hat in such a way that a bad action or let's say a suboptimal action ends up looking much better to begin with then the optimal action. You can get in a situation where you keep choosing the wrong action anyway and so you are only going to learn things that reinforce that action which might be good just not optimal. So you won't actually end up converging onto the true Q hat and that's why you get into a local min. That's right and so in that bad situation and you, I admit it's a little contrived because I've never, I've never even been to Chile is that it wont learn. It'd actually be exactly the same as this first case, always choose a sub 0. Mm-hm. So, that seems problematic too and it interacts in an interesting way with the initialization. Right. So maybe we can do this idea of using Q hat, but we have to be much more careful about how we initialize. Hm. So, you know, what I want to do is something like random restarts.

AS. Choosing Actions Two

So I think that is a clever idea, random research. That was one of the ways that we got unstuck when we were in local optima when we were doing optimization. Maybe this will also come in handy, in this setting. Yeah I like the idea because I

came up with it but I can see a couple, you know, but I can see a couple of problems with it. But well you can't see a problem with it yet because it's not even clear what you meant. Or maybe that's the first problem. Well it was clear what I meant in my head, now if you're going to want my mouth to understand what my head meant, well then your just asking for too much, so I really kind of meant, random restarts where you just kind of start it over, over, over, and over again, the problem with that, the problem with that is, it's already going to take a long time, you know, infinity to get to a good answer. And we thought we might have this issue with randomized optimization, certainly this is going to be a problem here, but it still feels like there's something we could do with the randomness idea that would help us to overcome this problem with using what we know. And only using what we know. Alright, alright, so I, I like that direction. So let's think a little bit about what that could mean. So random restarts was one idea that we had when we were talking in optimization about getting unstuck, which is, let yourself get stuck, and then once you realize you're stuck, throw everything out and start over again. Right. And you're right. That's going to end up being really slow, but we had a different way of Of using randomness to get unstuck. In, for example, algorithms like Simulating annealing That's what I was thinking about. Yeah, so Simulating annealing. The idea is Simulating annealing is that you tend to, you take up hill steps. But occassionally, you're willing to take a random down hill step. Right. So, it's kind of this mixture of choosing randomly. And using what you know, what seems to be the best. Right, and so, yeah, yeah, I can see that. So then the random restarts thing kind of works if instead of it being a random restart, it's, it's just a random action every once in a while. Yeah, excellent alright. So simulated annealing like-approach says we're going to take a random action sometimes. So then our exploration policy, our approximate policy, is going to be. To, we're in state s. Figure out the best action in that state according to our estimate and take that with probability I don't know. let's say one minus epsilon. Mm-hm. In otherwise take a random action and see what happens. I have a probability of epsilon. Yeah that's whats left over so. Right so that's why you had one minus epsilon because you want epsilon to be small. So epsilon is going to be. So every once in a while you will randomly act, oh and then that'll, that'll solve the sim annealing problem. Or do what sim annealing helps you with by just taking a random action in what looks like a bad direction, comparatively, sometimes. But it also means that you get to explore the whole space. And so you have a chance of actually learning the true Q. Exactly. So, and yet most of the time, you know, we're spending, assuming Epsilon is small, we're spending a lot of our time taking good actions, or actions that we think are good. But we're always holding out some probability of taking other actions to help us improve our Q hat. Mm, so if you do that infinitely long, and you've, this will, so this will let you visit SA. An infinite number of times, as long as epsilon is greater than zero. That's right and that the mdp is connected, right? So, there could be state action pairs, that just can't ever be reached, in which case you won't reach them. But that's okay, because since you can't reach them, they really don't matter. Yeah, they sort of don't exist. Exactly. Alright so let's let's focus in on this a little bit more because I think that this is now the first idea that we've had for choosing actions that has the property that it will learn and it will use what it learns. Mm, I like that.

AT. Greedy Exploration

Alright, so let's say something that we actually know about this approach to action selection which is called Epsilon Greedy Exploration. And what we know is that if the action selection if GLIE, which means Greedy in the limit with infinite exploration, and that basically means that we are decaying our epsilon for epsilon Greedy. That we start off more random and over time we get less and less random and more and more Greedy. Then we have two things that are true. One is that Q hat goes to Q, which is, which comes from kind of a standard Q learning convergence result. But, we also have something cooler in this case which is that the policy that we're following, pi hat, is getting more and more like the optimal policy over time. Hm. So, not only do we learn stuff, but we use it, too. And this is an example of the exploration exploitation dilemma. And exploration exploitation is really, really important, not just because they're two words that surprisingly start with the same five letters, like unlikely letters. But also, that it is strike, it is talking specifically about this issue. Exploitation is about using what you know. And exploration is about getting the data that you need so that you can learn. And this is one particular way of doing, it turns out there's lots of other ways of making this trade-off and the reason it's a trade-off is because there's only one of you. There's only one agent acting in the world and it has these two actually somewhat conflicting objectives. One is to take actions that it doesn't know so much about, so it can learn about them and the other one take, takes actions that it knows are good. So that it can get high reward. Hm, that makes sense. You know, I didn't, I never realized that exploration and exploitation share the first five letters. Hm. I always knew that they shared the last five letters. Oh, that's interesting too. Huh. So if you take an r and turn it into an it, you might move from exploration to exploitation. I feel like an entire political movement could be founded on that. [LAUGH]. But I'm not sure exactly what it would be yet. Maybe I'll work on that, before our next lesson. So I have an algorithm. There, there's a standard, lemma. Mm hm. In reinforcement learning theory called the exploration exploitation dilemma. Sorry, [LAUGH], no, lemma. The other kind of lemma. The exploration exploitation lemma, which has to do with taking actions that are either exploring or exploiting. But I have one where you actually do teaching. Mm-hm. You can actually, each time you take an action it's either going to teach the agent something, it's going to explore or exploit. So I call that exploration exploitation, or explore, exploit, explain. Oh, nice. But you could have called it the exploration exploitation dilemma because you used dilemma. And di means two sometimes. And you do the two things. Well, in fact, dilemma does literally mean two. Its a choice between two things. Right, so its a dilemma. Nice Its a lemma about two things. Alright, so there is, it turns out there's a lot of other approaches to exploration and exploitation. And some of them in the, in the model based setting. You can do a lot more with it, a lot more powerful things with it because you actually can keep track of what you've learned effectively in the environment and what you haven't, so the algorithm can actually know what it knows and can use that information to explore things that it doesn't know and exploit things that it does know. Q learning doesn't really have that distinction. It's a much harder thing to do. So, so that's what I wanted to tell you in terms of, you know, thinking about exploration-exploitation, does that make sense to you? It does make sense to me. I think what, the main point I got out of this, or a main point I got out of this, other than our incredible ability to get

caught up in letters and coincidences of spelling, is that the exploration-exploitation dilemma really is a dilemma. It's like the fundamental tradeoff in reinforcement learning. You have to exploit because you have to use what you've got, but you have to learn or otherwise you might not be able to exploit profitably. So you have to always trade off between these things, and if you don't, you're bound to either learn nothing or to get caught in local minima. I couldn't agree with you more. In some sense, if you think of reinforcement learning as being the question of model learning plus planning. There's nothing new here because model learning is well studied in the machine learning community and planning is well studied in the planning and scheduling community. Planning. And so, like, what are we adding to this? And what we're adding is the fact that these two processes interact with each other and depend on each other and that's exactly the exploration, exploitation dilemma and that's information has to go back and forth between these two processes that other people understand and we're the glue. Or the glee. [LAUGH] The glee glue. I like it. That's beautiful.

AU. What Have We Learned

So that's at least in a nut shell the reinforcement learning story. There's there's a lot of other topics and I think we are planning to get to some of them in a topics lesson a little bit later, but there's you know, courses and books worth of stuff to study in this area. Just like supervised learning as a whole. But, we're going to just kind of end it there with the idea that we now have a handle on how we can use learning to do decisions with delayed rewards. So, can you help us summarize what we what we learned in this lesson? Okay, sure. I think what did I learn here today, I think I learned a lot of things, some of which had to do with reinforcement learning Mainly that you can actually learn how to solve an MDP. I think that's actually a pretty big deal. Right. Meaning that we don't know T and R. Mm-hm. But we just have access to. The ability to interact with the environment and recieve transitions. And, that's actually that's actually pretty impressive. And, a very powerful thing. Because, we're often not, if we assume the world is an MDP, but we don't know TNR. If we don't have some way of learning in that we don't really have much we can do. And, you've showed me that their is something we can do. Cool. I think that's the biggest thing. We learned some specific things. In particular, we learned about Q-learning. Several kinds of Q-learning, but one is actually a real word. Yes [LAUGH] indeed they're all real, Michael. We learned about Q-learning. And I think the other most important thing that we learned about is the exploration versus exploitation trail. And with Q-learning, we learned a little bit about when it converges. Mm-hm. And that it's actually a family of algorithms. And different members of that family have different behaviors associated with them. Oh, there's one other thing I wanted to say on that topic, actually. Okay. Which is, that one way to achieve this exploration-exploitation balance, was to randomly choose actions. So to change the we're doing action selection. But there's another one too, which is that we can actually by manipulating the initialization of the Q function. We can actually get another kind of exploration, can you see how that might work? Oh, I know what you do. If you could, if you set, say the Q values to all be the highest possible value they could be. Great, so if we initialized the Q hat to awesome values, then what the Q lettering algorithm would do, even with greedy exploration, what it will do is it will try things that it hasn't tried very much, and it still thinks are awesome. And little by little, it gets a more realistic sense of how the environment works. And. So it's very optimistic? That's right, exactly and it's referred to often as optimism in the face of uncertainty and it's a similar kind of idea that's used in algorithms like, A*, if you're familiar with search algorithms in AI. Oh yes, I remember those. But this is, this is a really powerful idea and it's used in, in reinforcement learning and banded algorithms and planning and And search. Okay, and that makes sense because if everything is awesome. Then your true key value can only go down if awesome is bigger than the biggest Q value you could ever have. And so that means you're going to look at every single action. And as you learn more about them, then you will just get more depressed about them. And that's good. [LAUGH] Yes the world slowly beats you down. [LAUGH] So is that it? Is that all we really talked about? I guess that's about right. We talked about what a que function was. Right. And how that kind of binds everything together and we talked about different approaches including policy search and model base reinforcement learning. Yeah that was very nice. We tied it all back into planning. So one, one thing we didn't talk about is connecting to function approximation, and the issues in machine learning that are really important things things like over fitting. They come up in the reinforcement learning setting, but not in this simplified setting that were looking at here where we learn a separate value. For each state action pair, we're going to have to start generalizing to see the importance of that. And that's, we're going to do in a later lesson. Okay. I like it. And we also learned a bunch of things about letters. Like exploration versus exploitation. In fact, we know enough that we can now get an A in letters. [LAUGH] I like it. Okay. Well, I think we learned a lot, Michael. [LAUGH] Okay. Well, good. Well, thanks. It's very nice to get a chance to talk to you about this stuff. Cool. And so I guess. What are we going to talk about next. Well Whatever it says on the syllabus. I think it's game theory. That's pretty cool. Oh. I see why we're going to do that. Because all we've been talking about the world as if there were just one agent and nobody else. And now we're going to see what happens. When there are other people. Right. Other people show up at the party, next time. On, Machine Learning.

II. Game Theory

A. Game Theory

Hi Michael. Hey Charles. How you been? I've been doing just fine. How've you been? Good. It's been a while. I understand you went to Africa. I went to Africa. The entire continent, or at least parts of it. How's Africa? It's fine. It's, warm. The entire continent? [LAUGH] The entire continent. Or at least Kenya and Nigeria. Okay. Particularly in Nigeria. Which was 80 something degrees and 4,000% humidity. [LAUGH] So, do you want to do a shout out to any of, any of your friends from there? Yeah, I'd like to say hi to Chairman and I'd like to say hi to Prof. Oh, and Good Times. I would like to say hi to Good Times. He was awesome. Okay, so Michael. I am back now so that we can talk about the last little section. That we're going to do in this class, and it's game theory. That sounds cool. What does that have to do with machine learning? So that's an

interesting question because, in fact, game theory comes from a tradition way outside of machine learning in A.I. but you'll see in a moment why it is I care about game theory, and really this is a very natural extension to all the stuff we've been talking about with reinforcement learning. Interesting. Are we talking about games like Monopoly? In some sense, we are. Because all of life is like Monopoly. So in fact, what is game theory? Maybe that's what we can do. We can just try to define game theory, and maybe it'll be clear why it is we're worried about this at all. Seem fair? Yeah. Okay.

B. What Is Game Theory

Alright Michael, so there's lots of definitions of game theory that we could use. One that I like in particular is that game theory is the mathematics of conflict. Hm, [CROSSTALK] that's interesting. I think it's kind of interesting. Or generally it's the mathematics of conflicts of interest when trying to make optimal choices. because I feel like a lot of people have their own conflicts with mathematics. I think everyone but mathematicians have their conflicts with mathematics. I think that's fair. I see. But do you see if you, can you see how worrying about the mathematical conflict might be a sort of natural next thing to think about after you've learned a lot about reinforcement learning? I guess then well the next bullet kind of, kind of suggests a trend. So, so we've been talking about decision making and it's almost always in the context of a single agent that lives in a world and it's trying to maximize reward. But that's kind of a lonely way to think about things, so what if there's other agents in the world with you? Right and of course evidence suggests that there are in fact other agents in the world with you. And what we've been doing with reinforcement learning which, you know, has worked out very well for us, is we've been mostly pretending that those other agents are just a part of the environment. Somehow all the stuff that the other agents do is hidden inside of the transition model. But truthfully it probably makes sense if you want to make optimal decisions to try to take into account explicitly the desires and the goals of all the other agents in the world with you. Does that seem fair? Yeah. Right. So that's what game theory helps us to do and then at the very end I think we'll, we'll be able to tie what we're going to learn Directly back into the reinforcement learning that we've done and even into the Bellman equation. Oh, okay, nice. Yeah, so that is going to work out pretty well but, but we have to get there first and there's a lot of stuff that we have to do to get there. But right now what I want you to think about is this notion that, we're going to move from reinforcement learning world of single agents to a game theory world of multiple agents and tie it all back back together. It's a sort of general note that I think that, that's worthwhile is that, game theory sort of comes out of economics. And then in fact, if you think about multiple agents there being millions and millions of multiple agents, in some sense that's economics. Right? Economics is kind of the math, and the science, and the art of thinking about what happens when there are, lots, and lots, and lots, and lots of people with their own goals possibility conflicting, trying to work together to accomplish something, right. And so what game theory does, is it gives us mathematical tools to think about that. I feel like I feel like other fields would care about some of these things too, like sociology. Right. And what about, I could kind of imagine biology caring about these things, too. Even biology, I like the idea of biology. Biology. Why would biology care about this? Well, I guess the way you described it in terms of lots of individual agents that are interacting. Like, you know, creatures that live and reproduce. I feel like they, they have some of those same issues. Sure. So certainly biology at at the level of entities, at the level of mammals or level of insects, you might be able to think about it that way. But perhaps even at the level of genes and at the level of cells. Little virii and, and bacteria. You could possibly think about it that way. because they're in conflict too, I guess. Yeah. Now there's probably this notion of intention. It's not entirely clear what that means here and I think implicit in the notion of what we're doing here is this notion of intention and explicit goals as opposed to ones that are kind of built into your genes, but I think that's a perfectly reasonable way of thinking about it. I think the, the lesson from this discussion though is that. What game theory sort of captures for us or what would like for it to capture for us, is ways of thinking about what happens when you're not the only thing with intention in the world and how do you incorporate other goals from other people who might not have your best interest at heart or might have your best interest at heart. How do you make that work? And so if you think about that problem then I think it makes sense that it's been increasingly a part of AI over the years, and in some ways machine learning has started to think of it as being a mainstream part of what we do. Cool. Hence, why it's worth talking about today. Okay. Sound good. Gotcha. Okay. Let's, let's try to make this concrete with a very simple sort of example.

C. A Simple Game 1

Okay, Michael, so here's a, a nice little concrete example to, to think about this. Let's pretend that we're no longer in a world of a single agent like we've been thinking about with reinforcement learning, but we've gone full-blown generality to two agents, [LAUGH] okay? And let's call those agents a and b, and they're going to be in a very simple game where a gets to make a choice. And then b gets to make a choice. And then a might be able to make a choice. So this tree that I've drawn over on the right is going to capture the dynamics of this specific game that I'm imagining. So, these little nodes here, these circles represent states. And we can think about those in states in the same way that we. Talked about reinforcement learning in the past. The edges between these nodes represent actions that one could take. So, this should look familiar, this is just basically a game tree like anyone who's taken a, an AI course might have seen. Okay? I guess so. It doesn't look like a very interesting game. No. But I guess it's a, sort of abstract example. Yes. It's a very simple game just so that we can get a handle on some basic concepts. So, in particular, if you look at the details of this game, you start out in state one. Ok? And A gets to make a choice between two actions, going left or going right. If A goes right, goes right, she ends up in state three. If she goes left, she ends up in state two. Regardless B gets to make a choice. From state three we can choose to go right, and really that's all that can happen. And this, what happens if B goes right from state three is that, a value of plus two is assigned to A, okay? All of these numbers at the bottom, a the leaves here, are going to be values or rewards if you want to think about 'em that way that are assigned to player A. And, in fact, for the purposes of this game, it's going to be the case that B always get's the opposite of what A get's. So, if A get's plus 2 then B get's minus 2, if A get's plus 4 then B get's minus 4, if A get's

minus 1, B get's plus 1, does that make sense? Yeah, though could you write it down so that I won't forget? Okay, that's fine. So, by the way, this is a very specific type of game. here, and it has a name, which I want to get right. This is a two-player zero-sum finite deterministic game of perfect information. So as a, as a title or description of, of this kind of game, does this make sense to you? Do you think you know what they all mean, what all those words mean? So, two players because it's a and b, zero-sum. Because you said the leaves are a's rewards and b's reward is the negation so if you add the two rewards together you're always going to get zero. That's almost right. [LAUGH] Ok. It's not exactly right. Actually, so zero sum really just means that the sum of the rewards is always a constant. And that constant needs to be zero. It doesn't need to be zero. So if it added up to eleven, that would still be zero sum? If it added up to eleven everywhere. Yes. Huh, okay, interesting choice of terminology. finite, I don't know, everything seems to be finite here. There's no infinite number of choices or states or depth. Mhm. deterministic, well, again, thinking about it in an MVPish kind of way, there's no sort of casting transitions in this particular. Picture. Right. So if I'm in, state two and I go right, I always end up in state four, period. Right. Mm-hm. Game. I guess, a game is because it's more than one player? Sure. Of perfect information. That doesn't quite sound like the same terminology that we used in the empty MDP setting. But, I'm wondering if that's like, I know what state I'm in, when I'm making a decision. So, it's like a, like an MDP as apposed to a POMDP. Well it, that's exactly right. It's, it's that you know what state you're in and. Yeah. That's exactly what it means. It's like being the MDP versus the Palm DP. That's a great analogy. Cool. And does it matter that it's a tree like this? because when we were looking at MDPs, we had more complex structures of graphs and things. Well, you can think of this as unrolling the MDP if you want to. So then those states are sort of time stamped and history stamped. For, yeah, for the purposes of this discussion, yes. And that's a perfectly reasonable way of thinking about it. But, okay. But in general, we're going to be thinking about game trees, but actually, we're going through all of this for nothing, because we're going to discover pretty soon that none of this matters. [LAUGH] but, give me a couple of slides to get there, okay? [LAUGH] Sure. Okay. So this is about the simplest or at least the, the least complicated game that you can think about. Two players, zero sum. Finite deterministic game of perfect information. You know, basically, I can look at this tree, I know everything I need to know, and I can make decisions about what action I might want to take in order to maximize my reward. Okay? Good. All right. Now, in NBPs, of course we had this notion of. policies, right. You remember what a policy was Michael? Mapping from states to actions. So in the game theory world, we have something very similar to, policies. We call them strategies. So, all a strategy is, is a mapping of, [LAUGH] all, of all possible states to actions. So for example, here's a strategy, that A might have. When in state 1, go left. And when in state 4, also go left. Seems like a terrible strategy. Does it? Well, just, if nothing else, just in state 4. Sure, but it's a strategy, right? Okay, but it's just, it's a strategy for one of the players. Right, exactly, each player has a strategy. And that makes sense, right? Before when we talked about a policy, and mapped [UNKNOWN] to action, there was only ever one player, only ever one agent. And so we didn't have to worry about what other strategies there were. Here, when we talk about a strategy, it's always with respect to one of the players of the game. Okay, so, question. I've just given you one strategy, which is what A does in all the states A could potentially end up in. How many other strategies are there for A? For A? Okay, that sounds like a quiz. That does sound like a quiz. Let's make it a quiz.

D. A Simple Game 2 Question

Okay Michael, so here's the quiz, because you made me make it a quiz, I decided to make it even harder, so I want you to tell me how many different strategies are there for A and how many different strategies there are for B. And you are talking about just deterministic mappings only, right? Well, we are in a two player zero sum, finite, deterministic game of perfect information. So, [LAUGH]. So yes, I mean deterministic. Okay, alright. because, you know, it could be that it might be helpful to be stochastic. Oh, that's true, that's true. So in fact, that's a good point. I was going to mention that later, but I'll mention it now. What I just wrote down here, is actually not just a strategy, it's something called a pure strategy. Hm, it's always about purity with you. It is, purity of chocolate and bacon mainly. But, yes. So, these are simple pure strategies. Okay, so how many pure strategies are there for A and B. Okay. Okay, then let's go.

E. A Simple Game 2 Solution

Okay, Michael, you ready? Yeah. Alright, what's the answer? So I was thinking about A before you threw in the B. So let me, let me not think about B yet. I'll just think about A. So you had said in state one, it can go either left or right. And in state four, it can go either left or right. So boy, that sounds a lot like two times two equals four. Yes that's exactly right. But generally speaking, so actually walk me through that again Michael. How did you get two times two? So, well I, I had a little choice about how to think about it. One is that in some sense, if you go right from one, then you don't really have to make another choice. Right. But if you go left, then you have this other choice to make of either left or right. So it's, you if, if you're just writing it down as a mapping from state to action, you've got two choices at state one, and two choices at state four. Mmhmm. And so that is two times two, right? You can make, independently choose each of those. Right, that's exactly right. So in fact it's important there that even though if I can gone right on one, I would never, have to make another choice because I can't reach state four. In order to be a strategy, you have to basically say what you would do, in all states where you might end up. Okay, that's fair. Okay, alright so what about B, using that incredible reasoning? [LAUGH] So yeah, so B seems a little trickier because here it can only ever matter whether you're in like if player A sends us down to the left then we have a choice of three. If player one sends us down the right, we have a choice of one, which is really no choice at all. Mmhmm. You can have any color car you want as long as it's black. Yeah, like my Tesla. I was thinking the Model T, but maybe T stands for Tesla. T does stand for Tesla. So, by the definition of how many different you know, sort of reachable strategies it would be one answer but, if you're defining it the way you're defining it, it's going to be three times one or, three. Yeah. And that's exactly right. Good Michael. So, let's actually write down what those strategies are.

F. A Simple Game 3 Question

Okay, Michael so I have written out biased on upon your rather impressive way of thinking about it. All the possible strategies for a. And all the possible strategies for b. A, can go left, left, left right, right left, right, right. For states one and four respectively. And b can go Left right, middle right or right right. Right. Right. In particular, three in stage three we can only go R. Which makes me a? Conservative? A pirate. Oh. That makes more sense. It does make more sense. Okay. So, now why did I write it this way Michael? Well, I wrote it this way because if I write it this way with all the choices, all the strategies that b has up here and all the strategies that a have here. I actually form a matrix. And I can put in each of cells of the matrix the value of taking a particular from a and a particular strategy from b. Does that make sense? Yeah, that's very clever. Yes, it is very clever. I'm very happy that I came up with it entirely on my own. Okay, so let's start filling in these numbers. Or, instead of filling in the numbers ourselves, we could ask the students to do it by making it a quiz. Nice, I see. Shall we do that? Sure doesn't seem very hard. Okay, so let's make certain everyone here understands and what exactly we're asking you to do. We're saying that if for example A takes this first strategy go left and stay one and left and stayed four. And B takes it's first strategy which is go left and stay two and right and stayed three. What is the value for A that will result? So, let's actually do the first one as an example and ask everyone to do the rest of them. That seem fair? Yeah, that's what I was going to suggest too. Okay, good. So let's see. If A chooses to go left in state one, since A goes first, we'll end up in state 2, right? And then in this first strategy. B goes left in state two, which means we will end up going down this path and the value there is plus seven. So seven is in fact the value of this game with respect to A. Now we know that because this is a two player zero sum finite deterministic game of perfect information. That if a gets a value of seven, b gets a value of minus seven. So we could write minus seven here, seven comma minus seven here. But since we know that it's equal and opposite let's just write down the value for a. Okay? Just for compactness-sake. That seem fair to you? Yeah. Okay, cool. So with that in mind, let's see if we can fill out the rest of this table. Think you can do it? I think so. Okay. Then. Can you give me a place to write everything? It will magically appear. Go.

G. A Simple Game 3 Solution

Alright Michael. Tell me the answer. let's go across. Alright. Across. So. The first row it's all left, left, but the second left doesn't matter because that's in state four. Oh it might matter. I take it back. So left, left you did was seven. Left and then, player two goes to medium, middle. Middle. That would be three, so then the pay off is three. Mm hm. If the first player does left, left and the second player does right, right; then we're going to go left, right, left again, or for a minus one. Mm hm. Oh wait, hang on. Left, right, left. Yes, huh, yeah. The next row should be exactly the same as this one, except for in that third case. So it should be seven, three. And the r, the difference there. The r in four only matters in this case where we've gone to the right state in [INAUDIBLE], sorry, right action from state two. And so that should give us a plus four, so four. Correct so far. All right half way there, so, all right now, now, we've got something different so now player one is going to the right, so now actually player one's choice in state four never matters, so the next two rows are going to be identical to each other. And so in the first case, oh, actually [LAUGH] all of these numbers are going to be identical to each other, so player one goes right, player. Or, sorry, player A goes right, player B has to go right. There's no choice. So we're going to get 2s, no matter what now, so it's going to be two, two, two, and then the second row will repeat that two, two, two. Okay. So there you go. Michael, you are correct. You can feel very good about yourself. Very good, very good. So you're right, Michael, we do have we do have this nice little matrix and we have these numbers, and you know what's really interesting about this? That we can figure out what the payoffs are for B without having to do any additional work. That's true. But it's even more interesting than that. Which is, now that I've written down this matrix, nothing else matters. This whole tree, all these rules. None of it matters. It's irrelevant. Wait, wait, but. Okay. Because everything about the game is captured here in this matrix. This is actually called the matrix form of the game, and it comprises everything you need to know about it. It doesn't matter ha, what it means to go left or right or what it means to go middle or whatever. The point is by following this strategy and this strategy, you end up with seven for a. This strategy and this strategy, you end up with two for a, period. And how you got there does not matter. But it does creep me out a little bit that what your saying is that all that matters is the matrix. You know we should write a movie about that. I think it's been done and it's scary. Well, not the first one. It was kind of scary. Like people are trapped in this big box and their slurpyness. But their happy. But they don't know their happy. What do you mean they don't know their happy? [LAUGH] What is means to be, Michael, Michael, Michael. Speaking of which now how do we, how do we figure out. What do we do with this now? So that's the big question right? So, what was the whole point of doing reinforcement learning? The whole point of doing reinforcement learning was to optimize your long term expected reward right? Yeah. And so you pick the policy that would get you to the best place possible, so here's a question I have for you. Give that this is everything we need to know, this little matrix of values. These are all the policies A could choose from. We're calling them strategies here. But, you know, strategies, policies. There are four A can choose from. And there are three B can choose from. What will you think A and B actually do? Okay, so, you know? A wants the rewards to be high. So, seven is the highest, so A should choose the upper left corner. But A can't choose the upper left corner. Why? A can only choose a strategy. B gets to choose some strategy. I see. So A is choosing the row and then B gets to choose the column. Yeah. So then B should choose that also because that's what A wants. No, B wants to maximize what B gets. And remember, B always gets the opposite of A because. Because it's a two-player, zero-sum, finite game of perfect information, deterministic something. So, so you're saying that if we, if A chooses the first row. Say. Mm-hm. Which is the left, left strategy, and B now has a choice between three values and will choose the one that is worst for A, which would be the minus 1, so that would be a terrible thing to do. Yep. Okay then, so then what if A chooses the second row? Okay if A chooses the second row? So now again B is not going to let them have that seven which is kind of sad, but still cant make it to bad for A so B would choose the middle right. Strategy. Mm-hm. And then get, and then that would be a 3 for A, a minus 3 for B. But wait, hang

on. So that was done thinking of it as A move, A kind of making the choice first. Mm-hm. That doesn't seem fair. Okay, well then do it the other way. Alright, so if B chooses first, then B could choose the far right column because that's where the minus 1 is. Mm-hm. That's where it's going to be happiest. See, B seems kind of, kind of mean. It's like it's happiest when others are suffering. Well, but A is also happiest when others are suffering. I see. So, if B chooses that column then A wants to choose the second row and get the four, so B could choose the middle column, which then A would choose the two. A would choose what? One of the twos. One of the bottom two rows. No he wouldn't. Oh, right. A is trying to maximize, so A would choose one of the top two rows. Oh, yeah, that makes more sense. Right. Then choose a three which is better for B than having chosen the far right, and B could choose the far left column, but then A would choose one of the sevens, and B would be unhappy with that. So we'd end up with B choosing the middle column and A either choosing the top or the second row. Mhm. So that's kind of the same answer we got the other way. Yep. Huh. That's exactly right. And is that, was that just luck, or did you make this example to make that happen? No, I didn't make this example to make that happen. In fact, the process you went through is exactly the process you would expect to go through, and you will always end up with the, with the same answer. For a two player zero sum game. You know, deterministic, finite, all those other words. [LAUGH] The process you just went through, is exactly the process you would expect to go through. So let's see if we can, if we can be a little bit more, you know, explicit about the process you went through.

H. Minimax

So, in particular, the way I heard you write it down was that A must consider the sort of worst case counter strategy by B right? I see, because when A chooses the row, then B was going to make things bad for A along that row, so that's the counter strategy you mean. Right, and in fact, when you try to do it the other way with B. Well, B has to do the same thing. B has to consider the worst case counter, as well. And in this particular case, the way we set it up. Where the values for A. A is always therefore trying to maximize. And B is always trying to minimize A. Which works out to be the same thing as maximizing itself. Does that make sense? Yeah! I mean, other than the fact that, you know, I name my first child Max. I really wanted to name my second child Min. [LAUGH] That actually would have been pretty cool. Why didnt you do that? Because I'm married to someone with more sense then i have. Yeah i understand, i completely understand. Okay so, A is trying to maximize B is trying to minimize they both have to consider the worst case that the other will do. And so whats thats going to is going to force them through exactly the same process you went through. We just figure, I'm going to make the choice so that my opponent makes the counter choice, the worst case choice. I will end up as best as I can. So A is going to basically try to find the maximum minimum, and B is trying to find the minimum maximum. Hmm In fact that strategy has a name, or that way of thinking about it has a name. What do you think it's called? The minimum, maximum. Yes, or Mini max. Which is movie production company, I think. No, that was Miramax. Do you recall that where you have seen Mini max before Michael? In some other class that you once taught or once took years and years ago? Mmm. No. Mini max was exactly the algorythm that we use for game search. And intro to AI. Oh, which was a game tree, which is just what we started with in this case. Even though we turned it into a matrix. Exactly. So in the end, we, you know, this matrix induces a game tree, if you want to think about it that way. Or, game tree induces a matrix, if you want to think about it that way. And the strategy then, in sort of basic AI search, and the strategy in game theory is Minimax when you're in a two player zero sum game of perfect information. So is there a way to do alpha beta pruning on this? All alpha beta pruning does is gives you a more efficient way of finding the answer. I see but it's the same answer no matter how you set it up. Right. Cool. That's right. Okay, so this is pretty cool. So, we have set up a kind of game where we have multiple agents, you know, or at least two agents in this case, who have different strategies. And we actually sort of know, [SOUND], you know, sort of, in a world where it's a zero sum game, and you know the other person is trying to minimize what you get, maximize what they get. That the mini-max strategy would actually give you sort of an answer, in this case, by the way. We say that the value of this game for a is three. If a does the rational thing, and b does the rational thing, that is trying to maximize their own value, you will end up in this situation. That's kind of cool, don't you think? Very cool. I feel like there should be a theorem. There is in fact a theorem. I'm going to write it down.

I. Fundamental Result

Okay Michael, so here's this theorem that you, you so desperately wanted. You ready? Yep. I'm going to read it to you, because you can't read. In a two player zero-sum deterministic game of perfect information, minimax equals maximin. Alright you told us what minimax was, but you didn't tell us what maximin was. Well maximin is like minimax, except the other way around. So a is trying to. [LAUGH] You know, one side is trying to minimize the maximum, the other side is trying to maximize the minimum. Okay. It's exactly what we described before, just depends upon whether you're looking at it from a's point of view or b's point of view. Oh, I see, like, which, do you choose a column first or do you choose a row first? Exactly, so whether you go a first followed by b, or b first followed by a. You're end, going to end up in the same, with the same result. And that, more importantly, or at least as important, there always exists an optimal pure strategy for each player. In other words, you can solve those games and you know what the answer is. Once you write down the matrix. You just do Minimax, or you do Maximin and you end up with the proper answer. And now you know what the optimal players would do. Now there is a subtlety here which I got it a little bit in the previous slide, when I talked about rational agents. And what we're sort of assuming in everything that we discuss here is that people are always trying to maximize their rewards, okay? So we define the, the reinforcement learning problem that way. That my goal is to find a policy that maximizes my long term expected reward. You know so I'm trying to find, to get the best reward that I can. And what you're assuming here is that everyone else is doing the same thing and they're assuming that everyone else is doing the same thing. Okay. DOes that make sense? It does thought I'm a little bit stuck on this word optimal at the moment. Right. Well, that's what I'm trying to get at

actually. That optimal here really means I'm maximizing the reward that I can get, and I'm assuming everyone else is doing the same thing. And I'm, furthermore, I'm assuming that they're assuming that everyone else is doing the same thing. So, so I guess I'm wondering whether. Whether this theroem is vaccus in a sense that are we defining optimal to be mini max. What we're defining optimal to be, I think. So that's a good questino. I think I would unroll the vaccus one level by saying this. Optimal here, basically has to be optimal in respect to what? And the respect of what here is the underlying assumption that everyone is trying to maximize their rewards. And that everyone knows this. So, in a world where you have perfect information. It's zero-sum. Then, the strategy of Minimax and Maximin give you the same answer. And that furthermore there is always some place where the column and the row cross, the best column and the best row cross. And that is always going to be the solution to that particular game. Now, if we weren't in a two-player zero-sum deterministic game of perfect information, that might not be the case. But in a case where we're in this sort of simplest version, where everyone's being. Rational, that is, optimal, that is, trying to maximize their, their own reward. And assuming everyone else is maximizing their own reward, this is the right thing to do. Now, I've got a little weasel word here as well which we're going to get to in a moment which is this notion not just of a strategy but of a pure strategy. [INAUDIBLE] There's a reason why we have notions of pure strategies because in the end as we get more complicated we're going to have to do it with impure strategies. Mmm. Okay, but are you with me so far? I think so, yeah. So basically all that stuff we did in AI with game trees and search is kind of what you would expect people to do if they knew everything. [LAUGH] So, So, I feel like I could prove this theorum in the case of trees because you just prop, you just kind of commute values from leaves up to the root. Yeah. And, it, it doesn't matter. There is no notion of who goes first or who goes second. So there's just going to be one answer, to that process. It's not obvious to me, how to show it, if you, once you've. Turn the tree into a matrix, that that matrix, I guess because it captures the same information, it ought to be the case that this is still true, but like, I'd have to kind of sit down and think it through. No, and it's, so, so, to help you think it through, I guess what I would suggest is if you have the matrix, you can create a tree that's consistent with it. Because every row and every column represents a strategy. You don't know what that strategy is, but you can, since it's a finite matrix. You can construct a tree that is consistant with that major. In fact, possibly an infinate number of them, I'm not sure, but you can certainly construct at least one that is consistent with it. And then once you have the tree you just do what you said. Good. Alright, good. So we've got this fundamental result and now what we're going to do is we're going to try to be a bit more interesting. But it is important to go through this because now we've got some basic vocabulary and some basic building blocks okay? Yep. Alright.

J. Game Tree 1

So here's another game tree. We have, again, two players, a and b, and a gets to make a choice, first, to go left or right. And b will find herself in a situation, perhaps, where she gets to choose to go left or right as well. I've drawn these little square boxes, though, to represent chance. So what this means is that if a goes left, you end up in a chance box where you flip a coin and 50% of the time you end up over here, and 50% of the time you end up over here. Along a similar vein, if a goes right and then b goes left, then you end up in another chance node, and 80% of the time you end up here, and 20% of the time you end up here. Alternatively, if b goes right, you always end up here. Does that make sense? I think so. Uh-huh. So what we've done is we've gone from a two player zero-sum deterministic game of perfect information to a two player zero-sum non-deterministic game of perfect information. Okay, so we've relaxed, we gone at least one level up in complexity. Okay, so do you understand this tree? Do you understand this setup? Yeah, I think so. I mean here the stochasticness is happening essentially at the leaves. What does that mean? That there's no choices to make for either player after this randomness happens. But is that. That's not what you mean in general, right? No, that's right. There could be, after here, you end up in a different state where you can then make more choices. But because I don't have enough room, maybe because I want to make it simple, it just sort of ends after that. Okay. But yes, this tree could keep going on and there could be choice, random, choices chance nodes happening everywhere. There could have been a chance node that happened at the very beginning, even. I feel like I could work out the value of this game. Oh, well, how would you go about working out the value of the game? I would probably have it as a quiz. Okay, but what would the quiz look like? It would say, what's the value of this game? Okay. Do you think that just having that be the quiz is the best way for someone to learn or do you think maybe it can be done in stages? Oh, I don't know. Do you have other questions that you want to ask? Well, you know, how would you go about determining the value of the game? I think the first thing that you would do, at least if you were patterning after what we just did, is you would try to write down a matrix. Sure. So, I'm going to write down a matrix. So that our students will get a chance to, to work out what that matrix looks like and then from there figure out the value of the game. Are you okay with that? Sure. Okay. So if we were going to do that of course. The first thing we'd have to do is we'd have to figure out, figure out what the strategies are. So. What are a's strategies. I would have called them left and right but it, but they're not labeled. Yeah well let's do that we can call them left and right. A can go left or A can go right. What about B's strategies? B can go left or right. And B can go left or right.

K. Game Tree 2 Question

Here's your first quiz question. In a world where A can go left or right and B can go left or right. What are the values that you would put in the cells of this matrix? Again, this is zero sum. So, these values are from A's point of view and implicitly, there's we know what the values are for B. So, we're just looking for the values from A. Okay? Do you understand the quiz, Michael? Yep. Okay, so go.

L. Game Tree 2 Solution

All right Michael, you know the answer? Yep. Okay, what's the answer? Do you want to know the value of the whole game? No. I want to know the matrix. [LAUGH] Okay. So right. So, if A goes left, it doesn't matter what B does. And at

that point there's a chance node, and it's 50/50 for negative 20, which I feel like ought to be, negative eight. That's right. How'd you get negative eight? because half of 2 is sorry, half of 4 is 2 and half of -20 is -10. So -8. Right, so you just took the expectation of where you would end up. Exactly. Okay. What next? And that, it doesn't matter what b does. So then negative eight is also in the upper right corner of the matrix. Fair enough. The next easy on to do is if both go right. A goes right and b goes right then we get the three. It's just sitting there. Mm-hm. And the last thing requires me to do some multiplication. So Rrr. -5 times 0.8 Is like -4. Mm-hm. 10 times 0.2 is like 2 so -2. That is correct Michael. Shoo. So now we have the matrix. Now, what did we just notice here? Remember what I said about matrices before? It has all the information that you need. Right, so in fact none of this matters. Who cares how we got here? wait, but, but, oh. [LAUGH] I love erasing stuff! So here's the thing, we cannot reconstruct that tree from this matrix. We can't reconstruct, well, actually we could reconstruct that tree from the matrix. No, we can't. Yeah, we, of course we could. No, we can't. Yes, we can, because there's an infinite number of trees we could do, and one of them is that one. I see. We just don't happen to know that its the right one. I don't think that's really what people mean when they say I could reconstruct this. Like I could reconstruct the crime at this crime scene, it could be any of a million things. Like no, we can't construct that specific tree. But you know what it doesn't matter, because the only thing that matters is the matrix. Ahh. And you will notice, Michael, that you did all that multiplication and you multiplied 0.8 times 5 and after a couple of seconds of thinking about which was probably edited out but I know how long it took you do. You came up with the right answer. That's great. But notice that the number you came up with doesn't say anything about expected values and what the probablities are. It doesn't even matter that it's nondeterministic, because once you have these numbers, these expected values, that's what you're going to end up with. So who cares what the original tree was? Who cares if you can reconstruct it? Who cares what the rules of the games are? All you know is I have a choice of two strategies. We call them left and right here, but we could have called them one and two, or Q and Z. It doesn't matter. For the purpose of solving the game. So what is the solution for the game by the way? Oh, A is trying to maximize right? Yes. So A would never, ever, ever, ever want to go left. Mm-hm. So A's going to go right and then B is trying to minimize. So B is going to go left and we get the -2. I believe that is correct. And it makes sense the other way as well, right? That B would not ever choose to go right because then A would choose to go right as well. Gotcha. So you'll still end up here.

M. Von Neumann

So, here's in fact another theorem for you Michael. So, it turns out the theorem that we wrote down before is still true in the case of non-deterministic games. Oh, cool. Of perfect information, right? By the way, do you know whose theorem this is? Charles' theorem? No, although that would be cool. Von Neumann's theorem? Yes, this is von Neumann's theorem. Oh really? Yeah. Did you just guess? Von Neumann, he's responsible for everything. Do you know, well, you going to remind everyone who von Neumann is? He's my uncle? No. No, you're right. You're right. Von Neumann, so we talk about von Neumann architectures in computer science, that the basic design of a microprocessor is still following the ideas that that he worked out. Right, the von Neumann Machine. So there you go. So, this is pretty good, right? So, what we did, so what did we learn so far? What we learned so far is, the only thing that matters is the matrix. And once you have the matrix, you used mini max, at least if you're in a two player zero sum game of perfect information. At least if you're in a world of two player zero sum games of perfect information. We can just write down this matrix, throw everything else away and use mini max or maxi min and figure out what the value of the game is, which is the same thing as saying as, we know what the policy is. The kind of joint policy, in a world where everyone is being rational and trying to maximize their rewards and assumes everyone else is doing the same. That's pretty cool, I think. Awesome, so though, you know, I feel like I could make a matrix that wouldn't have this property. This maxi min equals mini max. And, and that we couldn't make a tree out of it. You probably could, but I'm going to guess that it's going to require that it's no longer zero sum. Nooo, I'm going to say no to that. Well, you know what, I, actually now that, I think I understand what you mean and the answer is yes you could. And in fact, that's what we're going to do next when we relax the next little bit. Ooh. So, would you like to do that? Sure, though I'm afraid if we get too relaxed, we might just fall asleep. In fact, why don't we do that, why don't we take a nap and then come back? [LAUGH] the, the joy of being on video, asynchronous napping. Done. Okay, I'll see you in a second.

N. Minipoker

Okay Michael, so here is a, another little game for us to play. And, what I want you to notice about this game before I describe it to you in detail is that we have relaxed yet another one of the constraints. So, we started out playing two player, zero sum, deterministic games of perfect information. There's also a finite in there somewhere. From now on we're just going to assume everything's finite, because why not? And then, what we did last time. Just a few seconds ago. Is we relax the deterministic part, so we have two player, zero sum, non-deterministic games of perfect information, and now we are going to relax the requirement for perfect information. So now we're going to look at two player, zero-sum, possibly non-deterministic games of hidden information. And this is really important Michael because, this last little bit of relaxation, going from perfect information to hidden information is going to be sort of a quantum leap into the difficult problems of game theory. So this is where it actually starts to get interesting so we have been building a foundation so far and now we are going to get to the interesting and complicated stuff, okay? Wow, one of the things that I learned just now is that the opposite of perfect is hidden. Yes. I always thought the opposite of perfect is imperfect but okay but hidden. I guess if I do not have a perfect face I should hide my face. Thats in fact the atomology of the phrase. Alright, I understand now. Yeah, cool, It's in wikipedia. Here we go, let me describe the game to you, are you ready? This is a version of mini poker where there are a set of cards, but they have no numbers of faces on them they are just red or black, okay. And red is bad, for our hero, Player A, and black is good for our hero, Player A. Okay? I see. So– Wait. Why's it have to be red? I don't know, man. You know, red. You know how it is. Okay. So, here are the rules. They're all written down on the screen, but let me walk through them with you. So A

is delta card. Magically. It will be red or black. and, the probability of it being red or black is 50% each. Okay? Yes. Right. So we have a uniform prior over, over the color. Now, remember red is bad for A and black is good for A. So, it's going to turn out without loss of generality, that if A gets a black card, A's definitely going to hold onto the card. Okay? Now A gets this card. B, player B, does not get to see the card. A can choose to either resign or to hold. If A resigns given a red card, then he looses 20 cents. Okay? Okay. Wait. So A's dealt red. A may resign if but only if red. Right. And then A loses 20 cents. Right. Okay. Alright, okay. Okay, so this is a betting game. It's not strange it makes perfect sense its sort of a metaphor for life. Now A can choose to hold instead, hold the card. Thus requiring B to do something. So if A holds the card B can either resign or can demand to see the card. Now if B resigns then A gets 10 cents. Regardless of the color of the card. Okay? Yep. That make sense? Yep. Okay. Now if B chooses to see the card, in fact demands to see the card. Then if the card is red, then A loses 40 cents. But if the card is black, then A gets 30 cents. And since we're betting, this means that whatever A wins, B loses and vice versa. That makes it zero sum. Got it. Okay, is this all make sense? Yeah, I don't know if I can hold all those different numbers in my head. But I, but the basic pattern of it is, that as you say Red is, red is bad, black is good. If A gets a bad card, A can essentially either fold Mm-hm. Right, resign? Or kind of bluff, like hey I've got this great card and then A and if B believes that the card is bad then and calls A or, or, or, and, and just folds then, then A gets, A wins that. But if B says no I think maybe you're bluffing and calls him, then everybody's rewards are more extreme, I guess. ¿Exactly. So it's just a version of poker. A weird version of poker. Simple version of poker. But a version of poker, none the less. There's a minor little detail here which isn't that important, but you know, notice it's written that A will A may resign if its red. Basically, A will never resign on a black card. Because it just doesn't make any sense. And so there's really, it just, there's not point in riding it out. Okay. Because black is always good, sort of, nothing bad can ever happen to A, if A gets a black card. So there's really sort of no point in riding anything out. But that's just a minor detail. Regardless these are the rules. Okay? Okay. You got it? Sure. I'm going to re-draw this as a game tree which might make it a little easier to keep all the rules in your head.

O. Minipoker Tree Question

Okay so Michaell here's the tree version, of what I just said. So remember I draw squares as chance nodes. And so, chance takes a, chance. And half the time we end up in a state, where I have, where A has a red card. And half the time I end up in a state where A has, or we end up in a state, where A has a black card. Okay? If A has the black card. Then B gets to choose whether to hold or not, so let me add a little bit of color for you since you wanted some color. This is the place where A gets to make a decision. Okay. Yep. And then this is the place where B gets to make a decision. Got it. Okay? So, A is going to either be in a red state or a black state. Only A knows this. B does not know this, does not know what state He is in. And so let's say A is in a black state. Well, A can only hold in that case. In part because it makes no sense to resign. And then B gets to decide whether to resign, and therefore A gets ten cents or to see, in which case A gets thirty cents. By contrast, when we're in the red state A can either hold or resign. If A resigns he loses 20 cents. If A holds then B gets to decide whether to resign and which case A gets 10 cents, or to see the card in which case A loses 40 cents and B of course gains 40 cents. So this is just a tree version of what I just wrote. Does that make sense? Okay, yeah I can see how this kind of captures the flow of information but I feel like there might be a missing constraint to it. So [INAUDIBLE] I don't know if there's actually a constraint in this thing but let me just point out something. And that is that B has no idea which of these two states his in, and its hidden information. And because B doesn't know what states he's in. He doesn't know whether resign over see. In particular B knew that he was in this left mode state then he would always say see. If B knew he was always in the rightmost state, then he would always resign. But he doesn't know which one, so it's not entirely clear what to do. Neat! Where did you get this game? This game is awesome! . I got this game like I got all of the examples that we're using today, from Andrew Moore. He's clever. He is clever. And I have shamelessly stolen... All of his examples and materials for this. But he said it was okay, by putting up slides in the world and saying, feel free to steal all of the stuff. You want to write his name so that people can credit him? Yeah, I'm going to write it at the very end. When we talk about. Okay. What we've learned. Awesome. Okay. So here's a question for you Michael. We know that I wrote down a bunch of words which describe the game, and then I made it a tree, because I could do that. And it makes it nice and easy to see what's going on. But now we know, that at least everything else we've done. We want to make a little matrix. And so we want to figure out what the value is for different strategies for A and B. So I'm going to assert, and I think it's pretty easy to see, I hope. That A basically has only two strategies. Either A is the type of person that resigns when a card is read. Or A is the type of person who holds, when a card is read. Agreed? Interesting, okay. Yeah. Right. So it really is a conditional policy, right? It's basically If red hold to resign, if black hold to resign, but your point is that black isn't really a choice, red is a choice, and there's only two choices. Okay I can see that as the two strategies sure. Right and of course this does say, this is, this does kind of say when you're in the black case you know you're going to hold. So you know what's going to happen. Ultimately B can either be the kind of person who resigns, whenever A holds or chooses to see, whenever A holds. Right? I see so, like in the previous trees. B would have four different strategies Resigner see in the left state, resigner see in the right state. Here there's this kind of extra connection, this sort of a quantum entanglement between these two states that it, that makes them have to be the same. So there really is just those two choices. That's exactly right, although, I probably wouldn't have used the phrase quantum entanglement. But yes, there's an entanglement, certainly. because we can't tell which state we're in. You're either going to resign or you're going to see. And you just don't know what else to do. Okay so A's a resigner or holder. B's a resigner or a seer. So the question is what numbers go in this matrix? And we're going to figure that out by having a quiz. I kind of saw that coming. Mm-hm, do you think you can figure this out? yeah, yeah. Sure. Sure. Yeah, I think so. Yeah, see, see, see. Yeah, I can do that. Okay, cool. So, let's go then. Go.

P. Minipoker Tree Solution

Okay Michael whats the answer? Let's start with resigner resigner. Alright, resigner resigner. Resign resigner diner. Alright, E, so resigner vs resigner. So resinger, if A is a resigner that means whenever A gets a red card A resigns. Yes. And that would be a negative 20. Yep. But that's not going to be the answer is it. Because A doesn't always, A doesn't always get a red card A sometimes gets a black card if A gets a black card [CROSSTALK] than a resigner, resigner that means B is going to resign and, and a plus 10 will happen. Mm-hm. So those are the two possibilities and they're equally likely, so it's a minus 10 divided by 2 which is minus 5. Right. You were correct, sir. Whew. Alright. Which one do you want to do next? Resigner seer. Okay. So again, oh, so the, yeah, good. This is a good choice, because now it's the same as argument as before, except for when we end up in that far right node, and that means minus 20 in half the cases, plus 30 in half the cases Which is a plus 10 divided by 2, or plus 5. That's exactly right. Well done Michael. Thanks. Okay which one next? So holder resigner. So holder reigner. That means when A, gets a card. A is going to hold the card. Mm-hm. And that's true, red or black. Yep. And then, then, it's going to be B's turn, and B is going to, oh, we're doing holder resigner. So, it's going to resign. Mm-hm. So, oh, well, interestingly, I think that takes us to those two leaves, both of which are plus ten. Yep. Because, why does that make sense? Because B, oh, 'cause B doesn't get any, no. Right, right, right, because it's independent of the card. You actually said that when you explained the rules. Mm-hm. So its, its average of plus 10 and plus 10, which ought to be plus 10. It is in fact plus 10. Well done. Okay what about holder's here? Whew, alright, so that's a case where A holds so we go down those branches and we end up, we always end up in one of the blue circled states. Yep. And B sees half the time that leads to minus 40, half the time that leads to plus 30. So that's minus 10 divided by 2, which is minus 5 again? Yeah, that's exactly right. So that's correct, Michael. And that's pretty cool, isn't it? Yeah, I really like this game. I didn't think you could have anything that had sort of poker essence and be this tiny. So yeah, so we live in this really nice little structure here. So I have a question for you. You ready for the question? I'm trying to guess what it is, but sure. Okay, here's my question. What is the value of this game? I was thinking that you might ask that. So, can I, can I step through it, is that okay? Yeah, sure, go ahead. So a's choosing the first row or the second row. So if a chooses the first row, and then b chooses the column, then if it's the first row, then b is going to choose the first column. So a's going to get minus 5. Mm-hm. The same story's going to go through on the bottom row. If a chooses the bottom row, then b's going to choose the seer position which gets the minus 5. Right. So from this so, it seems that the value of the games minus 5. But now let's do the same thing on the b column. So if V, resigns, now a gets to choose resign or holder. And it gets a plus 10. And if B is a sire. Then A chooses between re signer and holder and gets plus 5. Yes. So then from this perspective, the value of the game is plus 5. So, so here's a case where it better not be that we could take a perfect information game and put it into a matrix and get this out, because this is something that can't, like, it doesn't, it doesn't fit your theorem, right? We can't get the value of it by doing minimax or maximin. Exactly, the problem here is that once you move the hidden information. Minimax is not necessarily, and in this case, definitely is not equal to maximin. So von Neumann. Fails. As we all see. Idiot. Yeah, what has, what has he ever done for us, anyway? His theorems and his computer architecture that rules the world. Anyway, so we seem to have a problem here. And the problem is that once we go to this hidden information, as I promise complexity enters in Michael, and now we can't do something very simple with the matrix, find a pure strategy that's going to work. It really is the case that A strategy depends upon what B will do and B strategy depends upon what A will do. And if you don't already know what that's going to be, you don't actually have a value for the game. And in some sense you can get every single one of these values. So, there's got to be some [INAUDIBLE]. I feel like, that's sort of what you'd expect in a game like this. Right. So, if, because of the way that it is. If you know that I'm always a resigner. That I'm always going to, what? [LAUGH] Oh, that when ever I have a red card, I'm going to resign, then. You know that if I don't resign, I have a black card, so you know that you should resign. Mm-hm. Yeah, so it's, it's, it's one of these things where if I am really consistent and never bluff, say, then you can take advantage of me, and vice versa. Like, if you always respond the same way to when I act a certain way, then I can manipulate that. So it's kind of like this game of this sort of mind game like I want you to not know that I know the thing that you don't know that I don't know. Right but the problem is that I know what you know and I know that you know what I know. And that I know that you know that I know that you know that I know that you know that you know what I know. Did you really think that I didn't know that? [LAUGH] And so you end up in this terrible situation. But. There was a key word that you used there, Michael, and it was consistent. And everything we've talked about so far, pure strategies, is exactly the same thing as talking about consistency. So, the way we're going to get around this is, we're going to stop being consistent, or at least, consistent in the same way. Okay? Yeah. And to cheat here, is that we are now going to introduce, instead of pure strategies. Let's see the opposite of pure. Is? Impure. Mixed, that's exactly right. Contaminated. No, so, rather than sticking with pure strategies, we're going to start using mixed strategies.

Q. Mixed Strategy Question

So what's the difference between a pure strategy and a mixed strategy, Michael? Well, it's, it's simply this. A mixed strategy, simply implies, or means, some distribution, over strategies. So in the case of two strategies, like we have here, where you can either, A can be either a resigner or a holder, we're going to simply say that the mixed strategy for A is some value for P. Which is the probability of choosing to be a holder. So do you, you see what's going on here? So the only difference between a mixed strategy and a pure strategy. Is that for a mixed strategy, you choose some probability over all the different strategies that you might choose. So you decide that, you know, going into this I'm going to flip a coin. And half the time I'm going to be a resigner, and half the time I'm going to be a holder. Say. Or 30% of the time I'll be a resigner. And 70% of the time I'll be a holder. Okay? Yep. Where as with pure strategies, you always chose on or the other. So technically, it's the case that a pure strategy's also a mixed strategy where all the probability mass is on a single strategy. Makes sense. So in this case we're going to, in fact, choose P to represent the probability for A of choosing to be a holder rather than a resigner. And so

P can be 0%, probability zero, or can be probability one, or any value in between. You with me on that? Yeah, that's neat. Okay, good. To make certain you understand this I'm going to give you a little quiz. Which I have up here on the screen. You ready? Oh, I see it. [LAUGH] It's like those, those very square boxes. Yes. I didn't even realize what, what, what this could be about. It certainly couldn't be a quiz because Charles has never drawn a straight box in his life. I had drew those Michael. It took me 17 hours. Oh man, you are, you're committed to this and I appreciate that. I am committed to this. Okay, so, given that we have a mixed strategy, and we have a probability P of A being a holder, here's my question for you. In a world where B is a resigner, okay? B is always going to choose to resign. What is A's expected profit? To make it easy for you, I copied the matrix over here in the upper right hand corner. Wait, wait, wait. If B is always a resigner. B is always a resigner. Then, what's, and what is A? A is going to choose to be a holder with probability P. Oh, so you want this to be a function of P. Maybe. If it's, if it's, okay, sure, maybe [LAUGH]. It could, well, I mean, yes. It's a function of P, it just might be a constant function that ignores P. It could be, in principle. Okay, now, after you figure that out, I want you to decide, in a world where B is the seer, B always chooses to see the card. What would A's expected profit be, in a world where A will choose to be a holder with probability p. Okay? Hm. Got it? And that's going to be, oh yeah, that could be different because it's a it's a different strategy, though you know seering. Yes Like like if you're a resigner you're a resigner, but if you're a seer then what you are doing is seering Mm-hm. That would be an anagram of resigner. How do you see these things? I don't know. They're just there. I just, they words just kind of mix themselves up. Alright anyway, I'm, I think I am ready to do the function. I'm, I'm ready to stop looking at the letters. Okay? And go!

R. Mixed Strategy Solution

What's the answer? If B is the resigner, we don't really care about the other column anymore. Mhm. Then what's going to happen is A is mixing between resigning and holding. Yap. And probability P is a probability of being a holder, so whatever P, whenever that event happens it gets ten, and whenever one the opposite of it happens it gets minus five, so I want to say ten p plus one minus p five. Okay, is that your. Let me try that again. Is that your final answer? Ten p. Well I could simplify it. Okay. Well then say it to me again. I'll write it out here so it's easy for you to see. Okay and that is. My answer? That's fair. Do you want to simplify it? You don't have to. Sure, let me simplify it, so it's a minus minus p five and a ten p so that's fifteen P minus one. Minus what? One. Minus what? Five. Uhun, there you go. Thank you. That's correct. We would obviously accept either answer or any combination of those letters. That, [LAUGH] No, I think I might have to do this quiz over actually. No, not the one. I'm talking about either 15 p minus five or ten p minus one minus p times five. Or ten p plus one minus p times minus five. Or any other combination that we can get push car to actually Bothered to, you know. Check. Im, Implement or check. Okay. Well, that was pretty good. And this, of course, is exactly the expected profit. As you put it, P times A as a holder and P times or P percentage 1 minus P percentage A chooses to be a resigner. And so it's just the weighted average between those two values. So, Kent, let me just double check that. So, if P is 0, that means it never holds, it means it always resigns and it gets -5, so that's right and if P is 1, means it always holds, so it should get a +10. And 15 -5 is 10. So, boom! Yeah, it works. You used math there, very good. Okay, what about B? B. So the same story, except on the seer side. Mm-hm. So yeah, I might need that space again. So 5, oh I see, right, minus 5 times p. Mm-hm. Plus 5 times 1 minus p. Mm-hm. If we simplify that we get. There's a minus 5 and another minus 5. So we get minus 10 p. Mm-hm. Plus 1. Plus what? Plus 5. There you go. See you learned. Okay you want to check it? yeah. Oh, that's a good idea. So again if P is 1, then that means you're always which should be the minus 5 and if we put in a 1 there, we get 5 minus 10 is minus 5. And if P is 0, that means that we're always a resigner, and we should get a 5 for that. And yeah, so we zero out the negative 10 and we get the 5. Exactly. Now, it's not clear to me why we're playing this game. Oh, it is clear to me why we're playing this game, because we want to figure out something about. Strategy that is mixed. Right. So this is how well a mixed strategy does, but not against another mixed strategy. This is against two deterministic strategies. But is it, Michael? But is it? So I'm going to, oh, okay. I'm going to notice something here, which is that you, as you astutely pointed out earlier, we have two functions. Of P or to equations of P, and by the way, do you know what they are? They're lines. Sure, because it's just a, it's linear in P, so that's what linear means. Right, so what would happen you, do you think if we were to actually draw these lines? I think we'd have two lines. Yes, and what would those two lines look like? Let's take a look, shall we? Sure.

S. Lines Question

Okay, so what I've done here Michael, is I have drawn both of these lines. So here's the line 15 p minus 5. This is a case when b is a resigner. This is my best attempt at drawing it to scale. You start out minus 5, as you point out, and you end up with plus 10. And this is the line where b is a seer. That's minus ten b plus 5, so you start out with plus 5. And you end at minus 5. Okay? Cool. So, what do you notice about these two lines? They make an x? So they intersect. They do intersect. Can you tell where they intersect? How would I solve that? You'd have a quiz. Okay Michael, so here's a quiz. Where do the two lines intersect? Think you know how to figure it out? Yep. Okay. Then go.

T. Lines Solution

Okay Michael, what's the answer? I don't know, but I would, here's how I'd get it. I'd set the equations of the two lines equal and find the p where the, where those values are equal. Oh, okay. So here let's do that. And I feel like that's like 25 p on one side and ten on the other Okay. So like 10 over 25, which is like 2 over 5, which is, like, 2 over 5. Yes, which is also 0.4. So 0.4, that's exactly right, and it's, you should do it exactly the way you said. Set the two equations equal to one another, do simple algebra, and. So what would have happened if p ended up being, like, not a probability? Then they wouldn't have crossed inside. Good point.

U. Center Game

So by the way, here's a question for you. We can make it a quiz but I'm not going to make it a quiz. What is the value of the game at p equals 0.4? So I just plug it into those equations, and so negative 10 times 0.4 is like, negative 4 plus 5, is 1. So I'm going to say 1. Mm-hm. So the value here is in fact $0.01. And if you put it in the other equation you get the same thing. Yes, right, because they're equal there. That's actually the answer to the entire kit and kaboodle. If. Sorry, where were the kittens? If A chooses a mixed strategy, where with probability 0.4 he chooses to be a holder. Notice that it doesn't matter whether B is a resigner or B is a seer. You will end up here and there will be a value of plus 1 penny to A. On average. The expected value is. Yeah. plus 1. Neat! Now you might ask yourself, well what if B decides to be a smarty pants and also do a mixed strategy? What do you think would happen in that case? So of course it changes something, but it doesn't change the value of the game, because B, if B is a resigner, A is getting one on average. If B is a seer then A is getting one on average, and an, any average, any convex average of one and one is going to give us one. That's exactly right. So in fact if you think about it, if B tries to do a mixed strategy between these two lines. It's going to have to be for every single point between the two lines, somewhere the average is going to have to be somewhere in between there. No matter how you weight that average. So it's like a bow tie is this space of possible payoffs. Right and since for any let's if I just, If B decided to pick some, you know, some values such that it's in this region of the space. And it's going to end up somewhere, say between here that's fine, or here that's fine, or here that's fine, or here that's fine, or here that's fine. More importantly, right here it's going to have to be between the two lines. Where those two lines cross. So, no matter what mixed strategy B chooses. On average we will end up here, so the value of this game is the expected value of this game and it is, plus 1 for A. Okay so hang on why is that not there are other values that can be obtained. There is a minus 5 a plus 5, true. Why is it plus 1? Well its plus 1 here on average, the expected value of this game. Is plus 1. So you say that the strategy for A is to choose .4. Mm-hm. But why? Like, why is that, of all the different values. What [INAUDIBLE], just because the lines intersect. I don't understand what makes that a good idea for A. Like, it's not like it gets any additional payoff for having things intersect. Well, if a is going to choose a mixed strategy, and b is going to choose a mixed strategy. This is the mixed strategy for a, that guarantees an expected value of plus 1. Meanwhile, let's imagine this is the way you're going to set it up. So we've been kind of talking in general that a's going to choose the strategy; b's going to choose a strategy, and they're going to go. And that, because we know everyone's rational, you already know what strategy b's going to pick, and b already knows what strategy a's going to pick Right, because we made this assumption about everyone is trying to maximize their own utility, there own reward. So in a mixed strategy, if you know you're going to go for a mixed strategy, you have exactly the same situation. B can figure out well what is it that A should choose and A can figure out what is it that B should choose. So notice that in this particular version of the game, the way it's setup. Even if A announced beforehand to B. Listen I am going to choose to be a holder with probability 0.4. It doesn't matter what B chooses the expected value is this +1. Right? Yeah. But imagine if A said I am going to pick, I'm going to choose to be a holder with probability 1. Well then, what should b do? B should choose to seer. Right. It's exactly the situation we were in before. Okay, but here's the thing I'm having trouble wrapping my head around. So, it's not special that it's an intersection, what's special maybe, is it that, you know, if b is always. Okay; taking what you said before, that a is going to announce a strategy. So for anything that A can announce B is going to presumably do what's best for B which is to just minimize right? Mm-hm. So if you look at that triangle at the bottom. Mm-hm. If you think about that as those lines that V, upside down V shape. Yeah exactly that thing. If you think about that as being the payoffs, the payoff function for A, as a function of the probability P that it chooses to hold. Mm-hm. Then we've chosen the maximum. Right. Right, we're trying to find the, the probability for which A gets the highest expected value. And it's at that, it's at that peak there. Hm-hm. But notice something, Michael. For any case where the two lines cross, you're going to have a function of this form. Well, let's be sure of that, because that's, I wasn't seeing, I was thinking that that was not true. because basically in this case, we have a, a line of positive slope and a line of negative slope. Mm-hm. Right? So, what if we, we can have an intersection between two lines of negative slope. huh. So again, by doing the same exercise you were doing before, where you draw the, you sort of take the minimum of the two lines at all points. Yeah. You'll end up with this. Where do you pick it then? The far left, p equals 0. Yeah. But not the intersection in particular. No, that's true. Okay, alright, I just, wasn't quite getting that. But so, the intersection is special, Because in some cases that is where the max is. It can only I guess the max can only be in those three places, right, it could be far left far right or the intersection. There' s no other way to have a maximum of lines. Right. And b by the way there's another case like this where they never actually cross. Mm. Or like this where they never actually cross. Got it. Right, but in either case, right, it's always going to be those three points. It's going to be the extreme or where they cross. If they happen to never cross, then that point goes away and you just pick the maximum. So what you could, so, in fact, if you wanted to think of an algorithm to choose the right thing to do for A, you basically plot the lines. You take the min at every point and you find the maximum point. Oh I see, so you could just kind of discretized it almost. Yeah, and you're done. Seem reasonable? Well discretized seems problematic but like I get that we're trying to find the probability so that we maximize. Oh wait we maximize the minimum of the two other things. So it's like maximin again. Yeah. Except here, so, in fact, it is exactly min and max or max and min, except, oh, I guess, if you're thinking from A's point of view. It's min and max or max and min but in this case, there's this other parameter, which is the probability. Which is how you determine how you're doing the min and max. Got it, right. In this bigger space, it's min and max. I see, yeah, yeah, yeah, that makes sense. What about so, why is A the one who needs to be random? Why isn't B the one who needs to be random? It's, you end up with the same, you end up in the same place. Because again, remember, you're, you're doing this kind of optimal notion, where underneath all of that is this belief that both people are going to be rational. So the only reason to do this, to take the, you know, maximum minimum, is if you believe that B is always going to try to minimize what you do. But B's in exactly the same situation. Right, from B's point of view, That's what's happening. Huh. It's the same equations? Yeah, it just it gets turned around. That's actually an interesting exercise. Maybe that's the next homework assignment. Okay. So, did that all make sense to you? [LAUGH] [CROSSTALK] It looks a little green and scribbley, but yeah. Sure. That's cool. That's how

you know when you're done, Michael. When it's all green and scribbley. [LAUGH] Does this generalize to more than two options? It seems like it's going to get messy fast. Yes, it does. It generalizes to more than two options. Effectively now, you're just doing a max over n things, instead of two things. And you have to search for, there's possibly way more intersections to worry about. Yeah, but all you care about is the minimum. Think of it as you're always drawing the minimum function.

V. Snitch 1

Okay Michael, so, feels like we've relaxed almost everything we could relax except for one thing. We're now going to look at two player non zero sum. None possibly none deterministic games of hidden information. Cool. And that's going to turn out to be messy, but get us to the place where I've secretly been trying to get us all along. So, let me describe a game for you. Very carefully, and let's see where it leads us. Okay, here's the game, you've got two people. These two people are criminals. Oh no! Are they smooth criminals? One of them is. [LAUGH] [MUSIC] [LAUGH] Oh, you're terrible. Okay, so we have two people. They're criminals. Okay, and unfortunately they've both been captured by the cops. I have actually no idea how to draw that, but let's just try to draw that like this. The cops come along, and capture them both because they are suspected in a particular robbery. Okay? Hm. And they take them and put them both in jail. So those are jail bars. [LAUGH] Okay? But they don't just put them in jails. They put them in two separate jails. Oh no. Okay. And one cop goes to one criminal and says listen, here's the deal. We know you did it. Okay? We know you did. And your friend over there, he's currently singing, singing like a bird. And he, is going to pin it all on you. So this is your last chance to give us a little help and admit that you two did it. Or, that the other guy did it. You admit that the other guy does it, you say it was his fault, then we'll cut you a deal. Now to make it worse, the cop tells him that there's another cop over there. [LAUGH] Who's talking to the smooth criminal, and is offering him the same deal. Hm. And whoever, goes first, whoever pins it on the other guy first, gets to walk. Okay? Hm. So, if I can get the curly-head guy to defect, okay? Hm. Then, I am going to let him walk and he will spend zero months in jail. Okay. Okay. On the other hand, if the other guy defects, then he's going to get to spend zero time in jail. And since we've got all that we need. We got a confession. To get this guy to go to jail. He's going to go to jail for nine months. Okay? Negative nine months? Yeah. Oh, that's a cost in months. I see. Uh-huh. [CROSSTALK] Yeah that's a cost in months. Okay? So if, curly-head guy defects before the other guy does, then zero. He walks. He pays no cost, other than the cost he's already paid for a life of crime. [LAUGH] Now if he refuses to drop a dime on the other guy, but the smooth criminal decides to defect he's going to lose nine months. That make sense? And the other guy's going to walk. So he's getting the same deal, they're both getting the same deal. You got it? Yeah. All right, now. It's a little more complicated than that, all right? But I, I hope that you're getting all, keeping all of this in your head. It's a little more complicated than that. There's actually four choices here, not just two. It's not just a case of. I defect or the other guy defects. I defect, the other guy doesn't. I don't defect, the other guy does. Okay? You could both refuse to drop a dime on the other. You could cooperate, or you could both rat out the other. At the same exact moment? At the same exact moment. So, there's a big thick wall here. Curly guy doesn't know what smooth guy is doing, and smooth guy doesn't know what curly guy is doing. Right. So the question is, if you have a choice between either defecting or. Which means blaming the other guy? Which means blaming the other guy you defecting from your friendship. Oh. Or you can cooperate, that is be true to your friendship. I can, we both people can defect, both people can cooperate, one person can defect and the other person cooperate, so there's actually four different options there. I fee, I feel a matrix coming on. Yeah, I think there's a matrix coming on. So let's draw the matrix, because I think that makes it easier to see. But you have the background here, right? I think so, yeah. The key piece here is that each person has a choice of either defecting from their friendship or trying to cooperate. Keeping their mouth shut, and, they don't know what the other person is doing. So, what are all the costs here? What are the worst case things? Let's just draw it all out as a matrix.

W. Snitch 2

So let's just call the smooth guy A, because that's what we've been doing all along. And he can either choose to cooperate, or he can choose to defect. Now by the way, I'm saying cooperate and defect here because they're terms of art, not because these are the words I would have chosen. Okay? B can do the same thing. B can choose to cooperate or B can choose to defect. Now I've set up the game in a particular way here. The, the, the cops have set up the game in a particular way here. If B defects but A cooperates, then B gets to walk and A has to pay the price, nine months in jail. So, I'm going to put into this cell minus 9, 0. So, this means this is the value of this set of strategies for A, and this is the value for B. So the first number. In the pair is what A gets and the second number's what B gets. Okay? Got it. You with me? Yep. Okay, now notice we have to do this now because it's no longer zero sum. It's not going to always add up to a constant. Oh. Well, it is, so far it's a constant negative 9. That's right, oh and in fact it looks this way as well because I have a symmetric deal here. So if A defects and B cooperates, A gets to walk and B goes to jail for nine months. Okay? Yep. So right now, you're right, it's looking like a zero sum game, but it isn't. Because what happens if both A and B drop a dime on each other? Well, they both confessed. So it's a little good, it's a little bit better than having one of them, only one of them, confess. The deal that the DA is willing to give them is that both of them will spend six months in jail. Now these are just, these are just, the, the numbers that are a part of the game. This is, I'm not computing this from anywhere. This is just the way the, the cops have set this up. And I'm going to, now what's an interesting question here is what happens if both A and B keep their mouth shut? They both choose to cooperate? It turns out that it's not a perfect world because they were caught with a bunch of guns they didn't have permits for. So if neither one of them admits to robbing the bank, they're still going to do a little bit of time. But in this case, it's a small weapon's charge and so each only spends a month in jail. I see, so now it's definitely not zero sum. Mm-hm. Because we have a negative 2 there, a negative 12 there, a negative 9 and a negative 9. Right. Okay. Okay. So, you got the game, you understand it? Yeah, I think so. It's very simple. Now, just looking at this, what's the best possible outcome for the duo? So

if they cooperate with each other, then there's you know, one month later, they're back on the streets, back to their criminal activities. Or they're reformed. You never know. Sure. Back to their choice of whether to have criminal activities. The, the mutual defection one, that, where they both defect, it's either, well, 12 months or six months, depending on how you think about it, but they don't do nearly as well. And then the defect and cooperate, it seems like there's a lot of incarceration that will happen. Mm-hm. So, it feels like the best for the, for the two of them is to mutual cooperate. Cooperate, cooperate. Yeah. And that makes sense. This is sort of what you want to happen. Both of them keep their mouth shut. They do a little bit of time, but on average, they do pretty well. Right? Yep. So my question to you is, is that going to happen? Sure, why wouldn't it happen? Well, you tell me. If I know that you're going to cooperate. Let's say that I'm A, okay? And you're B, and I know that you're going to cooperate. What should I do? You should cooperate. Should I? You've chosen, you've chosen this column. All right? I see. Well, if you think of it that way then it's not a joint choice, but it's actually an individual choosing. Then you're better off defecting because then you get off scot-free. Yep. Then, but I go to jail for nine months. I could have a whole baby in that time. Usually takes closer to ten but sure. But you go to to jail for nine months, I don't. So if you are just that cold. I guess because you're a criminal. It doesn't matter. Remember, the matrix is everything. The value of doing this to me is 0 versus minus 1. Alright, that's, that's cold, man. I agree, but it's what the numbers tell you. What if it wasn't criminal? What if it was just, you know, the amount of money that I was going to win in a, in a mini poker game? Yeah, but it's a different kind of mini poker game, because we're both, because. [LAUGH] It's like you beat me. But, by beating me that way, you like, kill, nearly kill me. So? Are you saying that when? What do you mean so? Are you saying you always let me win whenever we play poker? No. Okay then. Because you're cold, is that what you're saying? No. Wait, what? Exactly, right. So the point, Michael, is that if I know you're going to cooperate, you're going to choose this column, then I should defect. Okay, alright, so, fine. So now I'm going to jail for nine months. Mm-hm, if you choose to cooperate. Aha. If, if is good. So what you're saying is, I could drop a dime on you. [LAUGH] You could. So, you could choose to defect, and if I knew you were going to choose to defect, what would I do? Well, you already, you already showed your colors, man. You're, you're defecting, so I'm just switching, I'm just saving myself three months by, by ratting you out. Yeah, so we would end up here. By the way, you know that this game is symmetric, right? no. You defected first. [LAUGH] No, here's the thing. Since the only thing I care about is maximizing my own reward, my own value. I'm going to do the thing that makes sense for me to do in this case which is if you cooperate, defect. If you defect, defect. But you would do the same thing because your whole goal here is to maximize your value. Yeah, well you don't know me like that. Yes I do because it's everything that's here in the matrix. This is the wonderful thing about game theory. All the stuff that you're concerned about is all inside the matrix. Remember the rules of the game don't matter. There's no prisoners here. There are no criminals. There's just, I get a dollar, I lose a dollar, or I lose zero dollars. I lose $9 or I lose $6. Or, for that matter, cents. I lose one penny or I lose no pennies. I lose nine pennies or I lose six pennies. It doesn't matter. This is the value of these particular strategies. And I'm going to want to defect if you cooperate. You'll notice if you defect, I'm also going to want to defect. In fact, if you look closely at this matrix, you should notice something, Michael. From A's point of view, when does it make sense for me to cooperate versus defect? You mean if you have a, some kind of probabilistic policy over cooperate and defect? No, no, not even that. Just in general, is there ever a time as A that I would rather cooperate than defect or vice versa? Oh, I see. So if I know you're going to cooperate, I should defect to, to save myself a month. But if I know you're going to defect, I should defect to save myself three months. Either way, I'm coming out ahead. Right. So in fact, let's take a look at this.

X. Snitch 3

If I look at the value for me as A, between cooperating and defecting in this first column. It's minus one versus zero, right? Which number is bigger? [LAUGH], zero. Right. If I look at it in this column, it's minus 9 or minus 6. Which number's bigger? Negative six. Right. In both cases, defecting is better than cooperating. So in fact this choice, this strategy, dominates the other. In other words it is always better for me to defect than cooperate. So I will never cooperate, ever. Because it's always better for me to defect. So fine. So I'll never cooperate ever. That'll show you. By exactly the same argument, minus one versus zero, minus nine versus minus six. Defecting is always better. B will never cooperate. What's the only thing that's left? Pain. Pain. This is called the Prisoner's Dilemma. Huh. You said it was simple, but it seems kind of evil. I didn't say it wasn't evil. Those things aren't opposite of one another. It's not like perfect versus hidden. Or pure versus mixed. It's not easy versus evil [LAUGH]. Evil's often actually easier. It's easy and evil. So we're in a little depressing place here, Michael. You claimed in the beginning, and I agreed with you, that this is sort of the best option, you want everyone to cooperate, because that's sort of what's best for the group. But because defecting dominates cooperating both for A and B you're going to end up here. At least if you reason about it that way. Yeah, I see that. Hence prisoner's dilemma. The only way to break this is if somehow we could communicate and collude with one another. So that we could guarantee that we would both choose to cooperate at the same time. So that wall? What about that wall? What if you put a like a Skype connection or a Google hangout? Well, if you were forced to say what you were going to say at the same time, or somehow be able to punish someone maybe for, you know not doing the right thing, then you might be able to make a different decision. But for this very simple version where I'm going to do it once, even if I could hear what you had to say, if one of us, whichever one of us went first, the second one would always be able to take advantage of that. I do find this kind of depressing. Yes, it's a dilemma. It's a true dilemma. So, this brings us to a more general sort of strategy. This whole notion of strict dominance, works in this case. But, you could imagine very complicated large matrces where it may not work. But it turns out there's a generalization of this notion of dominance. That works remarkably well. And that's what people tend to use to try to solve these kinds of games, to find out what the true value of a game is. So letme describe that for you, ok? Okay.

Y. A Beautiful Equilibrium 1

Alright Michael. I'm going to define a new concept for you. It is called the NASH EQUILIBRIUM. Nice. Okay, here's the set up. You have n players. So we move beyond simply two players. You have n players. And each player has strategies that it can chose from. So here I'm referring to them as, S1, these are all the strategies for player one. As two are all the strategies for player two up to S N all the players for strategy N. Got it? Yep. Okay so each of these are set, so im going to say that the particular strategies, S1 star S2 star S3 star, so on and so forth as N star that is a strategy that each of the N players has chosen. Is in a Nash equilibrium if and only if for each one of those strategies chosen by the n players it is the strategy that maximizes the utility for that particular player, it is the strategy that maximize the utility for that particular player Given all the other strategies that were chosen. Now I actually find that difficult to kind of work through, so let me try to say it a different way. Given that you have a set of strategies as one star, as two star, as three star, as n star. We know that they are in Nash equilibrium, if and only if. If you randomly chose one of the players, and gave them the chance to switch their strategy, they would have no reason to do it. Interesting, okay. So does that make sense? Yeah. Right. So an equilibrium right, just in general, the word, the word sort of makes sense, right, an equilibrium is at a place where everything is balanced. And in some sense, there's no reason for anything to move because they're in balance. So we set that a set of strategies are a Nash Equilibrium if no one person has any reason to change their strategy in a world where everyone else's strategy remains the same. Then you're, then you're in equilibrium, and in particular you're in a Nash equilibrium. Nash equilibrium. Okay good. Do you know who Nash is? Ogden Nash was a poet. Not that Nash. I think there was a TV series with a Nash in it. Yes there was. But you're thinking of John Nash. That's right The Nobel Prize winning person who was the, was featured in the movie A Beautiful Mind. Yep, that's exactly right. And it was, in fact, a Beautiful Equilibrium. Okay, so, there's the notion of a Nash Equilibrium. It was admitted by John Nash, if you believe the movie, in order to pick up women. Which I, you know, you know I never, I've never seen this movie. Oh really? Oh, I watched it and it was, it was surprising how unhelpful it was in explaining what a Nash Equilibrium was. [LAUGH] I'm not surprised. I hope this is helpful, though. So, it really is a kind of difficult concept to completely wrap your, your mind around, but what it really boils down to is, listen, if we all picked a bunch of strategies and we knew that one other person, one of us, we don't know who before hand, but we know that one of us Would have the opportunity to change their strategy after they see what everyone else's strategy is. We'd be in a nash equilibrium only if that person, whoever that person is, has no reason to change their strategy. Gotcha. So, so specifically you've made a distinction between strategies that were pure and strategies that were mixed. Right. And I guess I don't see in this case which kind we're talking about. Right. So, in this particular case, we're talking about. Pure strategies; however, exactly this wording works with mixed strategies. Oh, I see. So you could have a pure Nash equilibrium or a mixed Nash equilibrium. Right. And so, you could, each one of these, instead of choosing a particular strategy and saying they're a Nash equilibrium, you could talk about a probability dristribution over each of these sets of strategies and say those are Nash equilibrium if. No one would want to change their probability distribution. Got it. So this works for both pure and mixed. Alright, so you think you understand this? [LAUGH] sure. Good, we will test that.

Z. A Beautiful Equilibrium 2 Question

Okay, Michael. So you think you know everything, so here's a quiz. So, here are two matrices, matrix prisoner's dilemma and matrix bunch of numbers that look vaguely symmetric but aren't quite. So here's what I'm going to ask you do do. In each of these cases find the Nash equilibrium. Rut-roh. So do, you told me what it was but you didn't tell me how to find them so that seems kind of rude. It was implicit. It's intuitively obvious even to the most casual observer. Oh, well then, okay. So, and just to remind me now, each row and column is a choice for one of the players or the other player. Yeah. And the first number in the pair is the row player, or A's pay off and the second number is B's pay off. Yes. And we might need probability distributions, or we might not, depending on what it takes to be a Nash equilibrium. Right. okay, I don't [LAUGH] I'm not, it's not like I see the answer yet but it, but I either will be able to find it, or I won't. At least I understand the question. Okay. And by the way, just to make it easier for you there are pure Nash equilibria here, okay? Hm. Alright, so you ready? Yep. Just circle the one or underline it or whatever it is Pushcar does to make it so that you can answer this quiz. You ready? Totally. Go.

AA. A Beautiful Equilibrium 2 Solution

Alright Michael, you got the answer? Yea, I'm ready to try and figure it out. Alright let's go. Let's try the first one. I feel like the first one's going to go rather well. So we need a pair of strategies so that no player is. Motivated to switch. Mm-hm. And you told us they were pure strategies, so we actually have a nice algorithm for doing this, which is we could just check one of them with the definition. Mm-hm. But, but in the case of prisoner's dilemma, I think a natural place to start would be that minus 6, minus 6. Mm-hm. So let's say that A chooses the second row and B chooses the second column, let's see if that's a Nash Equilibrium. So both players need to be happy. So if, would A be happier switching? If A switched, it would be getting minus 9, which is worse. Mm-hm. So A is happy, where A is. And if B switches, B would be getting minus 9, so boom, Nash Equilibrium. Done. Now I didn't verify that other ones weren't a Nash Equilibrium, but you didn't say. Find all of the Nash Equilibrium. You just said find, well you did say find the. So you kind of implied that there's just one. Right. But actually that's true, but you don't have to check this because we already went through an exercise where we, where we knew the answer was minus 6, minus 6. And the way we did that was we noticed that for A defecting the second row is always better than this. Right? And then in particular this row strictly dominates this row. Right. Which implies that if I picked anything on this row I would rather move to the other row. And you can see it. Minus 1 0 is better. Minus 9, minus 6 is better. That's what it means to be strictly dominated. So, I'd never pick this row anyway. And this same argument for B for this column. So, you'll notice that by getting rid of the things that are strictly dominated, the only thing we're left with

is this. And, it turns out, in fact, to be a Nash Equilibrium. So, this is correct. So, you just told me how to do my job, which makes me little sad. Well, in this instance. Because I already had the answer to this. But maybe, maybe you were signalling to me how I might attack this next problem. Maybe. Maybe. So, A is choosing a row and gets the first number. So, is there a row, that, were they dominates all the other rows? It doesn't seem that way. Mh-hm But maybe, maybe, oh I could start with B, maybe B, there's a column that dominates all the other columns. But no, it looks like it's totally symmetrical. Yep, so strictly dominated doesn't necessarily help here, and by necessarily I mean doesn't. So that was kind of mean. Thank you. All right, so oh, but we have something else we could do. Yes. So, there is a, the largest number that anybody can get is 6. Mm-hm. And there's a play where both of them can get the 6. Yeah. So there's no way they're going to want to switch away from that, because everyone's getting there kind of maximum out of reward. So A bottom row, B right column, gets us Nash Equilibrium. And that is in fact correct, and you can see it because from here I would always, it would always be worse for me, and it would always worse for me. So, these are in fact the Nash Equilibrium, for these two problems. Cool. They've seemed easier than I was expecting. Mm-hm.

AB. A Beautiful Equilibrium 3

So here are three, fundamental theorems that come out of all of this work on Nash equilibrium. They're all in the screen in front of you. I'll read them for you in case you can't read my handwriting or if there's some typo you discover that forces me to erase some words and then write some new ones. Okay, are we ready? Alright, the first one is, in the n-player pure strategy game, if elimination of all strictly nominated strategies, eliminates all but one combination of strategies, then that combination is in fact the unique Nash equilibrium. So that's, that's what happened in prisoner's dilemma, we got rid of all but one option and that option had to be the unique Nash equilibrium. You say, eliminate all of them, do we, is it possible that things that we couldn't eliminate in one round, we could eliminate in the next round? Yeah it's possible, so in fact, this is done in an iterated fashion. You get rid of whatever you can eliminate and pretend they were never there. Because the truth is, no one would ever choose them. And then with what you have left, you just do it again. So it is possible. Although not in any examples that we did. Okay. The second one is and this, both of these I think are kind of Obvious. They sort of make sense anyway. At least to me. That any nash equilibrium will survive the iterated elimination of strictly dominated strategies. In other words if you get rid of things that are strictly dominated you will not accidentally get rid of nash equilibria in the process. And that makes sense because, if they're strictly dominated then if you ever end up there, you would want to leave. And therefore can't be a Nash. And therefore can't be a Nash equilibrium, that's right. So, those two things sort of make sense I think. The last one is true but isn't at least painfully obvious, at least not to me, and that is. If n is finite, that is you have a finite number of players, and for each of the set of strategies, that set of strategies is also finite, in other words you're still in a finite game, then there exists at least one Nash equilibrium which might involve mixed strategies. So you're saying there's always an Nash equilibrium? Yes, possibly mixed, for any finite gain. So, when you said I relaxed everything, I didn't actually relax the requirement that it be a finite gain. Fair enough. But you stopped writing finite, so it's sort of the same thing. Yeah, I agree. Because why would you play an infinite game? I don't know. Maybe you got a lot of time on you hands. Mm. That would be like an infinite jest. [LAUGH] Okay. So those are the, the main, results here. So, what did we learn? What we've learned is that with interesting games. We end up in these weird situations where we have to figure out how to solve the game and one really nice concept, is this notion of an equilibrium and in particular the notion of a Nash equilibrium. Cool. And there you go.

AC. The Two-Step

Let me just wrap up by doing a couple quick things. Earlier on when we were talking, you made a kind of offhand comment about, or maybe I made an offhand comment, about the fact that because we're doing this little prisoners dilemma thing, we have this sort of wall here, we're going to end up in this little bad place. Right? With the -6, -6. And there was this kind of notion that what if we could hear each other, would that make a difference? And the answer was, no not really, because who ever goes first is going to lose. And if you try to get people to go at the same time, well in some sense, that's the same thing as having a wall and you can't hear each other if you have to go at exactly the same time. Right, does that make sense? I didn't quite understand. Why does the first, person who goes first lose? Because if I find out whether you're going to cooperate or defect, then I will just then do the thing that makes sense. So if you do anything but defect, I will defect, and then I will win and you will lose. So whoever goes first runs the risk of the other person screwing them over. So. I see. Unless, unless that person defects. Right, but if that person defects, the other person's going to defect. Right, so, but you don't lose any more than you would have. That's true, and, but you still end up in this kind of unfortunate place. So, one question you might have there is, yeah, but that's just because you can only do this once. But what if you could do this twice? So, let's imagine were, going to be in this situation today and then were going to be in this situation tomorrow because, you know, you keep leading me astray. So, whatever happens today I can use as information, for what I might do tomorrow. So, if I decide to go ahead and cooperate this time. And you defect, and I know you're the kind of person who defects, and so when we're in this situation again, I am definitely going to defect. So maybe in that situation, it's in your best interest to go ahead and cooperate because, then we can keep coopoerating together, and in the end it sort of works out. So maybe this notion of not just playing prisoners' dilemma once, But playing prisoner's dilemma twice or three times, or four times, or five times, might come to a different result. Yeah, that seems plausible. I mean, I could imagine saying to you though this channel in the wall, something that says, you know, cooperate with me or, yeah, right, or I will, I will stop cooperating with you. And so, you get more reward if you, if you cooperate with me. Right. So, my question to you is, what does happen exactly in this set? Where I have two versions, two consecutive games of prisoner's dilemma to play. What am I going to do? Ok, well now I mean, we could turn this into a game, right? I mean, that's what you were teaching us how to do. Yeah. So you, so now the game has

four, well, I don't know. So one way to do it is to say A has a choice to make on the first step, and [COUGH] that could be cooperate or defect, and then the second step could be cooperate or defect, so there's four possibilities. Right. B has the same four possibilities. But maybe we can actually have A be responsive to B. So A has two things to do on the first step and then two things to do on the second step for each thing that B did on the first step, for a total of eight combinations. Right. You could do that. [CROSSTALK] So if we made an eight by eight matrix, we should be able to solve it. Right. So, that's pretty easy to do. You just kind of draw an eight by eight matrix in, you know, that many dimensions. And it, you know, you end up right here. I wasn't thinking about that many dimensions. I just thought, like the normal kind of matrix. But, okay. Oh, I see, I see, I see. I mean its eight by eight, so that's kind of a pain. Right. So, its 64 cells and, I guess, I guess we don't want to fill that out. Yeah, but you know. We could fill that out. But, I'm going to help you out here by pointing out You know what? It's not going to make a difference. Oh. So, here. Let's see if I can walk you through why it's not going to make a difference.

AD. 2Step2Furious

Let's imagine I have 20 of these games. Okay? Sure. So I'm going to play these games 20 times in a row. Now what you've been doing is you've been going forward. You've said, well, if I can get us the right thing or if I can basically threaten you and say. If you screw me over this time, I'll screw you over from now on. Then maybe I can get you to do the right thing and vice versa, since it's a symmetric game. And then we'll end up doing the right thing all along. And that makes a lot of sense, right? Yeah. If you're going forward. But what if you're going backward, Michael? Let's imagine we are doing [CROSSTALK]. What is amazing to me is that is exactly backwards what you just said. So, let's imagine we are doing these 20 dot dot dot dot, and now I'm on the 20th game, okay? What's going to happen on the 20th game? Well, I mean, I guess I could have built up trust in you from the previous games and think that you're going to cooperate with me, because we've been cooperating together. And then that would be the perfect time to drop a dime on you. Exactly. So, if this is the 20th game and we're in this situation, remember whatever value we've been adding up along the way, that's sunk cost. Right. The only thing that's left is the final game. And the final game looks like this, which means we're going to end up here. But guess what, Michael? The final one is determined. So since we already know the outcome of the final game, the only one that we, the next one that we can look at is the 19th game. Well, we already know what the outcome of the final game is. So this is effectively the last game. Oh. And what is the outcome of that going to be? It's going to be this and backwards, backwards, backwards, backwards proof by induction because the only that proves computer scientists know how to do is proof by induction. It turns out that we will always defect For what it's worth, those are the only proofs worth doing. But, okay. That's true. It's a fine point. Truth by induction and perhaps by conduction [LAUGH]. So, there you go Michael, even if I can play multiple, multiple games to try to build up trust, the truth is the Nash equilibrium, if I filled out that eight by eight matrix you wanted me to fill out. I would still end up in the same place where I would defect both times. Certainly if, yeah okay, so if you're going to be a jerk then, then I have to be a jerk, but yeah, I guess that's right. That's, that seems pretty horrible. It is. And by the way this is not just something I'm making up. This is another theorem that comes out of Nash equilibrium. That if you have an n repeated game then the solution is n repeated, Nash equilibrium. So whatever the Nash equilibrium is for the first game, the one version of the game, is the repeated Nash equilibrium for all versions of that game. Okay, wait a, hang on, hang on. So I. Okay, I buy that. Mm-hm. But couldn't you, well so, in a game that has more than one Nash equilibrium, couldn't we like, al, alternate? We could, and in fact you'll notice that in so far, I have not talked about the case where you have multiple Nash equilibrium. Because in fact, that's it's own problem. If I have one Nash equilibrium, then we know what's going to happen. If I have two Nash equilibria, which one do we choose? Now, what's important to know here, is that if we have two Nash Equilibria then that means that if you're in any one of them, you won't leave it. So it's not like you'll move from one Nash Equilibrium to another, sort of in the general case. Because the fact that it's an equilibrium means that if everyone else is fixed except one person, that person won't choose to move. But. I haven't said anything about how you would choose among multiple Nash equilibria. And that's beyond the scope of this class. Okay. But it is an active researcher and in fact some of my own students have done some work in thinking about. What it would mean to choose and actually the answer always boils down to, let's not worry about that. [LAUGH] All right. But I'm still kind of disturbed by this, right? So it seems like it's sort of saying, is if we knew the world was going to end tomorrow then we might as well be greedy bastards. And, in fact, if we know that the world is going to end at any particular time, which of course it will. We might as well be greedy bastards. That's right. So, be greedy bastards. Oh, no! Or at least, or at least be you know, Nash bastards. [LAUGH] We'll, the Nash is kind of greedy right, it's like I'm always taking an action that's always best for me. Yes, you are. No. Every single thing we've talked about today. Has always assumed that you're always doing what's best for you. The fact that it might also be best for someone else is neither here nor there, you're always doing what's best for you. Right, I get, I get that, but it's, it sounds like this argument is saying that we might as well be like that, like all the time and never really form, like never, never self-sacrifice even a little bit for greater advantage even to yourself. Well, that's not true. This, it's just all hidden in the utilities, once you've gone through all the what I'm going to do today, what I'm going to do tomorrow, you add it all up, whatever the right strategy is to take. That's the strategy that you take because it's already accounted for all the self-sacrfice you might do that then leads to later good deeds or later good outcome. So the wonderful thing about all of this, is that this matrix is everything you need to know. It captures the future, past and everything else, so everything you need to know is right here. And it's all been reduced to a bunch of pairs of numbers. But you did say something kind of interesting though, Michael, which I think is worth pointing out, which is that everything you say requires knowing when, at least in this sort of iterated version, requires knowing when the world is going to end. So an interesting question to ask would be, what would happen if I knew the world was going to end but I didn't know when. Would that change my behavior. It, it doesn't seem like that should make any difference because it's still going to end. If not today then tomorrow. Or if not tomorrow then the next day. True. But I bet yeah it does make a difference. I'm willing to go and think about that. Okay, why don't you go think

about it and let me know. Alright. I'll, I'll tell you about it in the next lecture then. Okay, I like that. Excellent. Okay, cool. So, let's move on. [LAUGH] Okay, fair enough.

AE. What Have We Learned

All right, Michael. So, I think that brings us to the end of what I wanted to talk about anyway. So, can you help me remember what it is that we've learned today? In particular, what you've learned today? Sure. So, I guess the first think I learned is that game theory can make you depressed. And, in fact, in particular that my friend Charles, given the opportunity. Would totally drop a dime on me just to save a month of incarceration. Yeah. Wait, no, no, no, I don't think you summarized that correctly. Game theory is not depressed. It's depressing. Oh, yeah. That's a good point, that's a good point. And Michael is not cruel. He is the victim of cruelty. I don't think so. Because you want to know what the secret here is, Michael? You've got a little matrix of numbers. Those number capture what's going on. The truth, Michael, is that, if we were in Prisoner's Dilemma. I would cooperate with you because my utility is not simply the number of months that I would spend in jail. But it's the number of months you also would spend in jail. Oo. Interesting. So if I, the best way to beat Prisoner's Dilemma, is to change the numbers. [LAUGH] I see. It's like the Kobayashi Maru of game theory. Exactly. So there's an interesting question for you right there, Michael. If I had prisoner's dilemma here, here, let's write it out so you can remember. If I had prisoner's dilemma here, we already know that we're going to end up here, because that's what the numbers tell us to do. But what we'd have to do is change the game. So how would you change the game in prisoner's dilemma? I see. So, if, if we're thinking about it in particular in terms of I care about how long you spend in jail. Maybe not as much as I care about how much I spend in jail. Like maybe half as much. Mm-hm. Then, the payments shift, right? Right. So, now we have like minus1 and a half, minus 1 and a half in the upper left hand corner. Mm-hm. Minus 9 comma minus 4.5, minus 4.5, minus 9 and minus 9, minus 9. So, yeah. So now that, that bottom right becomes a lot less attractive if we actually care about the other person. Right. Well that's, that's, that's, okay, I'm less depressed now. Except of course, that requires that you feel that way internally and that I feel that way internally. There's another way that you could change the game here. Which is, what happens to sniches in jail? They are rewarded. No. No they're not. They're punished. Yes. Oh. So, if your a part of the criminal fraternity, and you don't like prisoners dilemma, then what you have to do is to create a system Where the people who snitch get punished. So it's not just the months that they spend in jail, it's everything else that's going to happen to them if they drop a dime. So you're saying that minus 6, minus 6, ends up being worse? No, what I'm. No, the minus, wait, no, wait, what? Yeah. Oh, the zero ends up getting, oh I see the zero ends up being worse. Because even though your not in jail your going to get I don't know somehow thwarted or, or punished for your past behaviors. Accosted. Interesting. That's right. So that's what you have to do and that works not just with criminals but with the real world. Whenever your in this sort of situation like a prisoners dilemma, you can change the game by changing everyone's utilities. Like for example hiring police officer, police officers or hiring members of the mob to take care of everything. I see. So it almost seems like what you're talking about is a kind of inverse game theory, where if there's a particular behavior that I want to see. How do I set up the payments and rewards so that that behavior is encouraged. Right, and by the way that has a name, and it's called mechanism design. Mechanism design? Yes. I'm not sure I understand either of those words. [LAUGH] Well, that's where you're trying to set up the set of incentives, the mechanisms that you're using to pay people. You're trying to design them in such a way to get particular behavior. This is what a lot of economics is all about. This is what a lot of government is all about. Tax breaks for example, for mortgage interest, encourages you to buy a home, rather than rent a home. I see, by changing the payoff structure. Right. Oh that's neat. And so that's what we learned today. At least right now. [LAUGH] Okay. Alright. So, let's see. So, just to try to rattle off some of the other things. The whole notion of Game Theory. We talked about, especially the idea that. You can think about a game as a tree or you could represent it as a matrix. And, I believe you said, repeatedly, the matrix has everything. Is that how you said it? Or the matrix is all you need. Yip! Let's see. We talked about minimax and maximin. Mm-hm. We, we relaxed a bunch of constraint on games. Mm-hm. So we, we looked at both perfect and hidden information. Mm-hm. We looked at both zero sum and non zero sum. We learned a lot today. We looked at deterministic and [UNKNOWN] I would want to say, but you called it non-deterministic. And assuming that we can get rid of the first two bullet items that look like maybe they were jokes. I would suggest saying things like, we talked about what strategies are and that they come in different flavors. We talked about the evil prisoners dilemma game. Mm-hm. What else? You gotta give me more, otherwise it's [INAUDIBLE]. Oh, more, good point. Andrew Moore gave lots of really good examples that yes. Michael may be cruel, but Andrew Moore is awesome. He's more cool than me. Andrew Moor is very cool. All of the examples that we've used today, or almost all of the examples we used today actually come from Andrew Moore's slides. Andrew Moore is a professor at Carnegie Mellon or at least he was before he went off to Google. And is a really smart guy who cares very much about machine learning. And game theory and produced a bunch of slides that it turns out lots and lots of people use in their own courses. And his examples were so good for game theory. That I decided to coop them with his permission of course. He tells everyone that they may use them. And, in fact, we have pointers to the slides in the resources links and folders for all of you to look at. And I recommend that you do. Did we learn anything else, Michael? The only other thing that I would want to mention is NASH which is a concept that is nashtastic. It is nashtastic. There are by the way, I should mention briefly. Other kinds of equilibria concepts they're beyond the scope of this class. But there's a whole lot more to game theory as you might imagine. And sometimes when they ask you to in these situations where you can't do what you want to do, you end up in these prisoner's lemonade situations. Other kinds of equilibria can get you out of it. And I'm going to argue without explaining why that the way that they get around this is by introducing Other ways of doing various kinds of communication. and, in fact, I claim they're a particular part of mechanism design. But that's a topic for another day. Okay. Fair enough. Okay. Did we learn anything else, Michael? I don't know. That's what I was thinking about. I mean, that seems like a lot to absorb. And, the other thing is that repeated games, even the prisoner's dilemma kind of unravel if you know when the end. And I was going to look into what happens if you don't know when they're going to end. Okay. So, I guess,

that will be something that we will learn. Next time. Right, what we, what will we have will learned? Yes. Future past tense. [LAUGH] Alright, Michael, well, I think that's about it. At least my brain is full. So I will talk to you next time, and you get to lead what I believe is the last full lesson of the course. Oh, exciting. It is excuiting. Alright, well bye Michael. You have fun. I'll see you next time. Alright. Bye Charles. Bye.

III. Game Theory Continued

A. The Sequencing

Hello Charles. Hi Michael. How are you today? I'm doing just fine. How are you doing? Alright. I'm a little out of practice with this lecturing thing. So, I hope this goes well. I'm absolutely sure it will. So, today's lesson is continuing what you were talking about in terms of Game Theory. But, I'm going to be focusing in on what happens when you are worried about making decisions. With more than one player in a sequence. Which we started to get into at the end of your discussion but I'm going to go, more into it. Okay. I really like the logo by the way. Thanks very much. Yeah, it's a, it's a specially game theory logo. I like it very much, I like it very much. I will point out, however, that all sequels should be called the quickening. Yeah, I was going to go with the quickening. Or judgement day, but I didn't, didn't think that made any sense. Hm, that's a fair point.

B. Iterated Prisoners Dilemma

So let me take you back into what we were talking about last time. We were talking about the iterated prisoners dilemma, and here's the prisoners dilemma payoffs that you wrote down for us. You remember this? I vaguely remember this. And remember it was about these two criminals, Smooth and Curly. And they were, deciding whether to cooperate or defect against each other when, after they've been arrested. Right, and they defect, they always defect. They always defect, right. So in particular, we say well what happens if it's, if they have multiple rounds in which to interact, and so here they are, here are the two of them, and if they've got one round to live, we did an analysis and we indicated that there's really nothing they can do other than defect against each other. Mm-hm. It's irrational to do anything else. Yep. So we said, all right, well, what happens if we allow there to be more than one round? So now we've got two rounds. And what we realized was that if you got two rounds to go, then these players essentially, face a one-round game because, after this round, what they're going to do in the last round has already been determined. So it's almost as if that round doesn't really matter. There's nothing we can do now that's going to change what they're going to do in that last round, so it's sort of like there's just one round and we're going to defect again. Right, so life is terrible and everyone is out to get everyone else. Exactly, and not only is it for two rounds, but this same argument continues as you go three rounds or more. Oh, it's like a proof by induction. It's kind of like a proof by induction, yeah, well it's proof by ellipsis. Mm, that's my favorite kind of induction. Proof by ellipsis. [LAUGH] So the question then becomes, what happens if the, number of rounds left is unknown, right? So what we've realized is that if you know how many rounds are left, the whole thing comes unraveled and they're just going to defect forever. But we raised the issue of, what happens if the number of rounds left is unknown? Hm. And it seems like it shouldn't really make any difference because if it's say some finite number we just don't know what it is, then it seems like it should still reduce to this same setup that we have. So I was looking into this and it turns out that is it's not the case it actually does make a difference and it's, it's really interesting how it goes and how it connects back with other things we've talked about. Woo, tell me more.

C. Uncertain End

So here's how I started to think about it. So let's say how can we represent the idea that we have an uncertain ending. We'll one way would be if we had some kind of generic probability distributions over the number of rounds that the games going to be played. But this seems like an, the simplest idea that I could think of. So here, here we have our, two criminals, and what their going to do is their going to play a round of prisoner's dilemma. But at the end of that round, they're going to flip a coin. And with probability one minus gamma, that will be the last round, it's all over. But with probability gamma, they're actually going to play again. And, and they do this after each round. And so each round is, is basically statistically independent of the other rounds. I see. So the set up here is, with probability gamma, the game continues. Now, notice that I chose gamma here. This was a, it's representing a probability here, but in the past, we've used this to represent a discount factor. Mm-hm. But that actually is the same thing, right? In, in, in the normal discounting, we say that the value that you get one step from now, is discounted downward by gamma. And that's exactly what you'd take if you worked out the expected value of a game where you continued making steps with probability gamma. And with probability one minus gamma, it ends if you get zero from then until the rest of time. So every round here could be your last, or not, right? It could be that you actually get to continue playing. Does that make some sense? That makes perfect sense. Awesome. All right, so, so yeah, this is, this is exactly that kind of situation where, well, here, let me, let me ask you a question. What's the expected number of rounds of this game? Well, I'll bet it's finite if gamma's less than one Yes, I even wrote that down. Yeah, I'm smart. Or at least I can read. Sure, but what's the, but, specifically we could actually write it as a, as a, function of gamma. Let's see. If gamma were something like, 99% then I would expect it to be about a 100, right? I think that's right. Yeah. So is that, is that your answer? [LAUGH] My function of gamma is if gamma is .99 the answer is 100. Yeah, something like that. It's not a total function but it's a function. Well, it's a sample. I mean, you do machine learning. Why don't you tell me what the function would be given that sample. Well, we can make it a quiz or I could just tell you. Why don't you just tell me. Alright. So one over one minus gamma is the answer. It works for your example. Um-hm. 1 minus .99 is 100th and we're talking 1 over that so, you get a 100. And, yea we could go through the argument as to why that's that's what it is. But this one over one minus gamma is what shows up all the time. If gamma is zero, then we're talking about one over one. The game lasts one round.

That's exactly what we'd expect. Mm-hm As gamma gets closer and closer to one this pro, this quantities getting closer and closer to infinity. So, Right. in fact if you know, it becomes unbounded as gamma hits one. So yeah. So this is the expected of rounds, and so that means like yeah. So as you said if gamma is 0.99, it's a 100 rounds. And we already, reasoned that at a 100 rounds the whole thing falls apart. Right, huh, and I noticed the one over one minus gamma, of course, is just like the way we did discount factors, when we started doing MVP's in the first place. Exactly, yeah, that, that kind of links them together. Hm, that's actually kind of neat.

D. Tit-for-Tat 1

So if we're going to be talking about strategies in this game that has an uncertain ending, we can't just write down sequences of, of actions anymore. We can't just say cooperate, defect, defect, defect, defect. Or even some kind of tree of possibilities. Because those are going to be finite representations. We need some other representation that allows us to play for an unbounded number of rounds. Mm-hm. And I'm going to start off by presenting an example of such a strategy, one that's, that's very famous for the iterated prisoner's dilemma, and it's called tit for tat. And the structure of tit for tat goes like this, on the first round of the game, A player playing this strategy will cooperate, and then in all future rounds, the player is going to copy the opponents previous move. Does that make sense? It does. So basically, we start, I start out, acting as if, you're going to cooperate with me. And the moment you don't cooperate with me, I will start to defect, and we'll be in the, the old style prisoners' dilemma. Right? Well no, not. What this says is that it actually copies the opponent's previous move. So if, if an opponent goes cooperate, defect, cooperate, defect, cooperate, defect, defect, defect, defect, cooperate, cooperate, cooperate. You're going to see something very similar coming out of the tit for tat agent. I see, I see, I see. In fact, we can represent the strategy as a little finite state machine, like this. Yeah, I like that, okay. And you, so you can see exactly how it kind of proceeds. It starts off cooperating. And then in each, each round it waits to see what the opponent does. That's the green letters here. And then it follows the corresponding arrow, to determine whether it is going to cooperate or defect in the, in the current round. Sure, that makes sense. So in this picture, the black letters here represent my move. And the green letters represent my observation of the opponent's move. Or at least if I'm being tit for tat.

E. Tit-for-Tat 2 Question

What happens if we follow Tit for Tat? That's a great question. So, let's make it a little more concrete. So, here's a set of strategies that an opponent might adopt. And now the question is, what does it look like Tit for Tat is doing in response to each of these? And I was hoping that you'd, you know, check the corresponding boxes. So basically, for each row, say okay, if you're playing against, if Tit for Tat is playing against always cooperate, what does Tit for Tat do? Does it always defect, does it always cooperate, does it cooperate and then defect, defect, defect, defect, does it alternate between cooperate and defect? And this should, just to give you some practice in, in interpreting the behavior of Tit for Tat. Okay, that works for me. I think I could do this. Go.

F. Tit-for-Tat 2 Solution

Alright, so let's start off, maybe you can tell me what happens when Tit for Tat plays against always cooperate. So what happens if you always cooperate? Well, let's see, I start out cooperating. Mm-hm. And since the other person is cooperating I will continue to cooperate because that's what they did the last time. Mm-hm. So, I will always cooperate. That's right. Good. Alright, so what about if we play against always defect. Well if we always defect, the first time I'm going to cooperate because that's what you said Tit for Tat is. Hm. But from that point on, I will do what my opponent does, which is defect. So I will cooperate and then defect, defect, defect, defect, defect, defect, ellipses. Yeah, so I put this always defect in there, but actually it can't ever be the right answer right [LAUGH] because Tit for Tat always starts off cooperating. So the, the other three seem like they might be possible, always defect is not possible. Good. Alright. So, what if Tit for Tat plays against another Tit for Tat? Well, I started out cooperating my opponent cooperated. And since I'm going to do what that person does I will cooperate, but since that person's doing what I did, they will also cooperate. And so we will both cooperate forever, so we will always cooperate. Nice. So isn't that kind of interesting? So Tit for Tat even though, if, when it's playing against itself, is a very cooperative fellow. Hm, I like that. But if Tit for Tat is playing against something that defects, it becomes a little bit more vengeful. Yes. Alright, so what if it plays against something that is a little unsure of itself? So it starts off defecting, then cooperating, then defecting, it's sort of almost an anti kind of thing, right? So it's, this is one that starts off with a defect. Right, so it always, so I, first thing I do is cooperate. And then after that, effectively I do what the opponent does one step before. So I basically take what you, I'm pointing to the screen, you can't see me. I take the the D-C-D-C-D, and I just put a C in front of it, because that's what I'm going to do. So, I will do, C-D-C-D-C-D-C-D, ellipsis. So that's your last choice. Isn't the C-D-C-D-C-D, ellipsis in Atlanta? It is, actually. It is the home of all such analysis of diseases in the United States. The Center for Disease Control? Yes. Is that what it's called? Yes, that's exactly what it is. Come to Atlanta and work for us. [LAUGH] Alright, so good. So, that, that's the pattern of, of responses that Tit for Tat makes against this set of strategies. Oh, so we answered my question.

G. Facing TfT Question

Alright so now that we have a sense of what tit for tat does against various strategies, let's try to think about what we should do against tit for tat, so what do we do if we're facing tit for tat. So I'm just going to break it down to two possibilities it turns out there's actually more, but these these two are pretty instructive. So let's pretend that we have to choose between always defect as a way of playing against tit for tat, or we have to be always cooperate playing against tit for tat. So what

I've written down here is what the total discounted reward is going to be or the total reward in this case. As a function of gamma. So, let's start with what happens if you play always cooperating against tit for tat. Well, you already told me that it. Such a thing will result in tit for tat always cooperating. Mm Hm. And that means we're going to play in this box. The cooperate-cooperate box. And that means on every single round, we're going to to get a minus one. Which means over an infinite run we're going to get an average of one, sorry minus one over one minus Gamma. Mm Hm. That makes sense. Okay? You agree with that? I do. Just minus one repeated over and over again. Now always defect as you recall you told me that, that will result, well for the always defect agent against tit for tat, the first thing that is going to happen is it's going to defect while the tit for tat cooperates right? So we're going to get zero for playing that strategy on the first round. Zero doesn't sound very good but look at the alternatives. They're all negative so zero's pretty good. Mm-hm. So it does this sort of you know good thing in the first step then after that tit for tat responds by always defecting in response. Right? And that means we're going to be stuck in this box the defect defect box where you get minus sixes for the rest of ever. Yes. So that means minus six over one minus gamma. But that starts one step from now so we multiply it by another gamma. Okay. Alright. So these are two different expression that represents what our pay off would be for adopting two different strategies. And in fact, if gamma is really high, very close to 1, then this is a really good answer, right? Because it sort of grows, it's like, minus over one minus gamma. So, for high gamma, we're talking about something that's minus one times a really big number. Mm-hm. Whereas this first one is not so good for high gamma because what's going to happen is it's going to get, end up getting minus six on every step. So it's going to do worse overall. But if, if we're talking about the low gamma. Then, let's say, you know, zero for example. A gamma of zero will, will always defect. We'll get zero plus zero. But always cooperate, we'll get negative one over one. And zero is better than minus one. So for really small gamma, like if the games unlikely to last many rounds, you should defect. But is the game is going to last a long time, then you should cooperate. I believe that. Cool! Alright, so then my question to you is: What's the value of gamma for which these two different strategies to play against tit for tat are equally good. I think I know the answer. Woah! That was fast. Let's give everybody else a chance to think about it. Okay. Go.

H. Facing TfT Solution

Alright Charles, what, you said you had the answer how do we figure it out? It's 1 6th. I guess that's what, which, we don't, I don't know if it will accept that, whatever 1 6th is expressed as a decimal, but yes 1 6th. How did you get that? I saw the number six and figured it had to be 1 6th because you said it was low, but here is what actually I did, well, I was thinking about it is, you said, well when are they equally good? Well, if you always defect, you get minus 6 gamma over 1 minus gamma. And if you always cooperate, you get minus 1 over minus 1 minus gamma, so they're equally good when those two values are the same. Good, alright, and the denominators are the same, as long as gamma's not one, that's fine. Right. If we divide by the negative 6, we get gamma equals 1 6th. Exactly. Excellent. So, so that's interesting, right? So, that's, it's saying that for gamma values that are less than 1 6th, we should be doi, we should just defect because there's no. Well the games not going to last long enough for us to form any kind of coalition. But for things higher than 1 6th. A half. 3 quarters. 0.999. It's going to be better to cooperate than to defect against tit for tat. Or 6 plus epsilon. Indeed. Yeah. I like that. That's actually very cool.

I. Finite State Strategy

Now, we kind of cheated here. Because I told you there's just these, those two strategies. But there's actually a bunch of other strategies you can play against tit for tat. And it's worth thinking through, how do you compute a best response to some finite-state strategy? So tit for tat is a finite-state strategy in that it has these two [LAUGH], these two states. And the strategies expressed in terms of transitions between those two states. But in general, if we have some kind of finite-state strategy, like tit for tat, how do we figure out how to maximize our own reward in the face of playing against that strategy? So in this picture here that I drew, the states are labeled with the opponent's choice, the finite state strategies choice, okay? Mm-hm. The edges, that's in black, the edges and labeled in green here, are labeled with our choice. So, for example, if we're in this state of the, sorry, if our opponent is in this state. Mm-hm. We have a choice. We can either cooperate or defect. On this round. Mm-hm. So, the green arrows tell us how that will impact the state of the opponent. And then these red numbers, I just added the information about well I know that if the opponant is about to cooperate and I choose to cooperate. I can just look up in the pay off matrix that that's a -1 for me. Right? Agreed? Agreed. So I just annotated all these edges, all these choices with these extra numbers. So, one of the things that's cool about this is unlike just the payoff matrix representation that we had before, our choice, it impacts the payoff, which is the same as that, but it also impacts the future decisions of the opponent. And that gives us this structure here and also says that maybe this is a slightly harder thing to figure out because of the fact that we can't just maximize our, the number. We actually have to think about where that's going to lead us in the future as well. So two things then. One, I was always fond of saying that the matrix was all that you needed. But that really only made sense when you were just playing once. Yes. That's right. Right? And, two, I look at this and it's a finite state machine but you know what else it looks like to me? It looks like an MDP. Excellent. It is indeed an MDP. Now, it's a, in this case, my opponent's finite state strategy is deterministic, so it's a deterministic MDP, but it is. It's a discounted MDP. Gamma's playing the role of the discount factor. The entries from the payoff matrix are playing the roles of rewards Our action is playing the choice of our action, and the opponent's internal state structure is playing the role our states. So it is, it's an MDP, and so how do we figure out what an optimal strategy is against a finite state strategy? We solve the MDP. Yeah, exactly. So any, any method for solving an MDP can then be used to actually compute the strategy, so what is the strategy going to look like? It is going to be a mapping from states of the opponent to action choices for us. Right, but that's fine because a state does not have to be your state, it's just what matters. What matters in this case is what they opponent is going to do. Right. So now what are the

strategies that can be meaningful against tit for tat? So if we cooperate then we're going to stay in this state and it's always going to be the right thing to do to cooperate. So always cooperate is one. If we always, if we defect, now we have a choice again so we could defect from this state which would cause us to defect forever, so always defect is another one. But what's the other thing that could happen? Well, we could tit for tat ourselves. Well, sort of. I mean, so, we could defect, so we could defect against cooperate, but cooperate against defect. Which would actually cause us to do D-C, D-C, D-C. So those are the only, oh I see. No, you're right. I'm not sure how to say it. But the policy is, defect when you're in this state, and cooperate when you're in this state. But the effect of that is to go back and forth against tit for tat. Right. Basically, take this loop here. And those are the only policies that matter. And in this case, we worked out that. Always cooperate is good against tit for tat if it has a high discount factor and always defect is better if you have a low discount factor. But, we can get that for real by solving the MDP. Right and that makes sense and the reason that those are only 3, let me see if I get this right, the reason those are the only 3 that makes sense because if you think of this as an MDP then it has no history, so when you are in C there are only 2 choices and when you are in D there are really only two choices so if you look at the way you've drawn it. You either stay were you are in c or d, or you take the loop. And those are really the only three options because the rest of them would require you remember what you did, you know, a time step or two ago, and there's no way to do that in an MDP, at least not as you've written it. Well, there's a way to do it, it just would never be better than doing it this way. So an MDP always has a mark off deterministic optimal policy. Right. So we only need to consider those. I think that was the same thing that I was trying to say, but with different words. [LAUGH] Ok.

J. Best Responses in IPD Question

Alright so, now that we have a handle on what it means to compute a best response against some finite state strategy. Let's actually take a look at these. So, so this is a quiz. Imagine that we have a gamma that is large, so greater than a six. Mm-hm. What is the best response to each of these strategies? So this will be a quiz, but let me just make it clear what the, the goal of the quiz is. So, if you're, if you are playing against, always cooperate. Which of these is the best response to it, which of these is going to actually have maximum reward? If we're playing against always defect which is going to have maximum reward? And if we're playing against tit for tat which are going to have maximum reward? Any questions about that, does that make sense? I'm just making certain I have the rows and the columns right. So my opponent is playing on the rows. Yep. And so I'm choosing among the columns for each one of those rows. Yes? Yeah. So I labelled it best possible response for the columns. And these are the opponent's strategies on the rows. Okay. Okay, go.

K. Best Responses in IPD Solution

Alright. So is this, is it clear what these answers are? Let's find out. So, let's see. we, we already worked out the math on this. So we know that, for gamma, greater than 1 6th, cooperating is better than defecting, in general. So if I'm going against someone who's always going to cooperate, then I should always cooperate. Incorrect. What? Yes, so, so, we didn't actually work that out. What we worked out was what to do if you were playing against tit for tat. If you were playing against someone who was always cooperating, and is completely oblivious to us. No, no, no, no, no, you're right. You should always defect because you're always going to win. Yes, yes, yes. You're always going to win. You're going to get zero on every time step. Right, right. Yeah. No, that makes sense. That's beautiful. Alright, what about always defect? Well, if you're always going to defect, you might as well defect. Indeed. because we're just in the regular old prisoner's dilemma world. [LAUGH] good, alright. So now we, now we have this, this other strange beast here. So our opponent is playing tit for tat. So we could always defect. Right. But we would do, we'd get a higher score if we can convince tit for tat to cooperate with us. For a gamma greater than a 6th. That's right. So that you should always cooperate. That is true, however. Mm-hm. What if we played tit for tat against tit for tat. You'll end up in the same place. Yeah, so that's just as good. Mm-hm. And that's, that's kind of interesting. If you think about mutual best responses. Yes. So that's a strategy that, a pair of strategies where each is a best response to the other, there's a, we have another name for that. Do you remember? No. [LAUGH] You taught, you told us what it was. Yeah, but I probably used different words. It's, it's a Nash equilibrium. Oh. This, that's what a Nash equilibrium is. A pair of strategies where each is a best response to the other. Each, ea there's no way that either would prefer to switch to something else to get higher reward. And that makes perfect sense. You're in an equilibrium. And that is a Nash equilibrium. Okay. So we can use this little table here to actually identify Nash equilibrium. So, what would a strategy be? So, so if one player plays always cooperate, then the best response to that is always defect. Mm-hm. But the best response to always defect, is always defect. So always cooperate is not part of a Nash equilibrium. Right. But what about always defect versus always defect? No, it is. So that's a Nash. Right? Yep. Because they're both doing the thing that is the best response to the other. Right. Alright, this box here, always cooperate against tit for tat. That's not okay, because if a player does always cooperate, it's always better to switch to always defect. Yep. But, check this out. If you are playing tit for tat, and the other player's playing tit for tat, there's no reason to switch because it's actually a best response, it's the optimal thing to do. And that works from both players' perspectives. And that makes sense. So like you said check this out and you've been using check marks that's very good. [LAUGH]. So we're in this situation where we have two Nash equilibria. Indeed, and one of these Nash equilibria, [NOISE] is cooperative. Which is the thing that we were sad about, or at least that I was feeling really sad about, in the, in the last lesson. The idea that, man, there's just, it's clear that they should just try to get along. You explained that you can modify the reward structure, and then they would get along better. But here it turns out, well, no, another thing you can do is just open it up to the possibility of playing multiple rounds, as long as you don't know how many rounds, it becomes possible to to have a strategy that is best off cooperating, and is in fact a Nash Equilibrium. Isn't that equivalent to changing the reward structure? It's definitely related to changing the reward structure. It's a particular way of changing it. A very particular way. But it's. Yeah, because it's not true any more that we can do this in the one shot case.

You have to be in the, in the repeated game, setting. Right, so you change your rewards structure to be a sum of rewards. But it's actually an expected sum of rewards, and you don't know where it is you're going to stop. So I guess you're changed, you're changed the game, you changed the, the rewards. But in a, sort of very subtle way. Yeah, the whole game is different, really. Yeah man, you changed the game. [LAUGH]. Sometimes you gotta change the game. Don't hate the game. No, you are supposed to hate the game. Don't hate the player, hate the game.

L. Folk Theorem

I'm really proud of us for figuring out a way to make the prisoner's dilemma a little more friendly. It turns out, though, that this, this idea, this sort of core idea, is very general and kind of cool. So, it leads us to a topic that we could call, repeated games, and the folk theorem. Mm-hm. So the general idea here is that when you're in the repeated game setting, this possibility of retaliation, the possible of, you know, not being cooperative actually opens the door for cooperation. Because now it becomes, it can become better to just get along and cooperate than to get retaliated against. Right. And that makes sense as long as the retaliation is plausible. Well, we're not talking about plausibility of the retaliation. We're just talking about, right, because the tit for tat, it's, it's not analyzed in that way. It, it doesn't say whether or not it's actually plausible. It just says, well, if you have this strategy and you play against this strategy, there's no incentive to switch. Right. But we'll, yeah, we'll get into this plausibility thing in a little bit. Oh, I, I stumbled across a word. Okay, go. [LAUGH] But I want to point out one thing first, which actually kind of irritates me. Kind of along the same lines as like, regression is the wrong word and reinforcement is the wrong word. Folk theorem is the wrong word. So, what is a folk theorem. It's two words. In general mathematics, sorry say again? A folk theorem is two words. That's fair, yes. So, I think it's actually two different terms of art. So in mathematics folk theorems are results that are known, at least to experts in the field, and they're considered to have established status but they're not really published in their complete form. There isn't some kind of original publication saying, oh look, I, I found this thing. It's more, something that like, kind of everybody knows, and so you don't really give anybody credit for it. So it's like a theorem that is in the general mm, cloud of understanding. It's sort of among the population, among the group. Wait! Does, does anybody every prove these folk theorems? You can, yeah! All folk theorems are provable. But it's not like you say you know this is Charles Isabel's theorem. Mm. It's like. No, it's just a folk theorem. It's like. Charles, Charles can prove it. We can all prove it. But we, we're not really sure whoever proved it first. It was just one of those things that everybody knows. Oh, so it's like an oral tradition. Yeah, yeah. I think that's a good way to think about it. Okay. I'm just imagining mathematicians sitting around a fire in the winter, cuddled up against one another sharing theorems that have been proved since the beginning of time to one another. Exactly. Right that's the image that I have as well. This, just it's folk theorems. It's this sort of, you know, I learned it from my grandmother and now I'm telling you. Hm. I like that. I like to think of mathematicians that way. [LAUGH] As grandmothers. Yes. So so that's what a folk theorem is in mathematics. However, in game theory it means something different. It is referring to a folk theorem, but it's also referring to a particular folk theorem. So the folk theorem in game theory refers to a particular result that describes the set of payoffs that can result from Nash strategies in repeated games. So it's a funny thing, it's a funny way that they use the word. It's, it's a, it's a folk theorem but it's also the folk theorem. Hm, well, well, maybe it was actually, proven the first time by some guy named Folk. That's a good idea, we should probably, we should push that forward like call him Folke or something like that. Folke's theorem Yeah. But but no. [LAUGH] Oh well, I tried. Yeah that's that's really great idea that we're going to just move on from. [LAUGH] But what I, what I'd like to do next is kind of build up to this notion of the Folk Theorem and it's going to require a couple basic concepts that are not too different from stuff we've already talked about, but we're going to kind of make them concrete and then we're going to show how they all come together to provide us with this idea of a Folk Theorem. Okay.

M. Repeated Games 1

The thing that I find most useful in trying to understand the folk theorem and, and what it says and how it works is a thing that I call a two-player plot. I don't know if other people have other names for it, but this, this concept is out and, and often discussed. I just don't know what it's called [LAUGH]. But here's the idea of it, it's a really simple idea. It's like a folk plot. It is a, it is a folk, a folk plot, right, I don't know who invented this plot. So here's what we're going to do, remember this prisoner's dilemma game. We've got two players, there's Smoove and Curly. Mm-hm. And, what we're going to do, is we're, there's a bunch of joint, actions that they can take. So Smoove cooperates and Curly defects, or they both defect, or they both cooperate, or one defects defect cooperate in the other direction. So what we're going to do is for each of those joint outcomes, each of those joint action choices. I'm going to plot a dot, put a dot on a two-dimensional plot, where this is the Smoove axis, and this is the Curly axis, okay. Okay. So cooperate-cooperate, remember from the prisoner, prisoner dilemma payoffs, is minus one, minus one. So I put a dot at minus one, minus one. Defect, defect. Is it minus six, minus six? Cooperate defect is it minus nine zero? And defect cooperate, is it zero minus nine? . So do those four points make sense to you? Do you understand like the idea? They do. It may not be so obvious why we do, do it this way, because as you have told us, the matrix is all you need. Mm-hm. But this is really just representing the matrix in another form. But it actually is sort of losing some information. Because these dots don't tell us what the relationship is in terms of if, if one player keeps the same action and the other player changes to a different action. The matrix captures that but this plot doesn't anymore, it's kind of washed out. I see.

N. Repeated Games 2 Question

All right. Now so just to kind of reinforce this idea and also to let us think about it in a slightly different direction, I want you to consider solving the following quiz. Here's four payoffs that I've put little boxes around. This one at minus 1, minus

1, this one at minus 3, 0, and this one at minus 4, minus 4. And the question is, which of these payoffs can be the average payoff for some joint strategy? And this is in the repeated game setting, [CROSSTALK] okay. So what we're imagining is these two players can coordinate in any way they want. We're not talking about trying to maximize reward or, or, being Nash or anything like that. They're just going to execute some, some pattern of, of strategies over infinite run. We're going to average the payoffs that the two players get and we'll say, you know is it possible for them to adopt a strategy so that the average per time step gets minus 3 for Smooth and 0 for Curly? Yes or no? So check that box if that's possible. Can it be that they both get minus one on average? If so, check this box. And can it be that they both get minus 4 on average? Check this box. Does that, does that make sense? It does make sense. All right, so I'll give you a chance to think it through and answer it. Okay.

O. Repeated Games 2 Solution

Alright, so are any of these easy to answer right off the bat? Well, one of them is really easy to answer right off the bat, and that's minus 1 minus 1. Good. Because that's one of the points, so. [LAUGH] Yeah. So, the joint strategy to get the minus one minus one would be for both players to cooperate, right? Again, they're just, you know, they're just willing to do that just for the sake of showing they can make that value. Right, and I think I know the answers for the other ones as well Alright, hit me. So here's how i'm going to, here's how I'm going to, going to talk you through my reasoning so that if any point I'm wrong, you can gently steer me away from embarrassing myself. Basically I was looking at these points and I thought hm, they form a kind of a convex hull. Aha! And I thought well, surely that's just an accident of the numbers and then I thought, oh wait, of course we're talking about averages here. So that means that all sort of possibilities have to be inside the convex hull of the outer points. So if I drew a line between those four points, I would end up with all possible, achievable averages. What, achievable averages, how? How would you achieve things inside the convex hull? I would appropriately average them. So in particular, the first thing I'd notice that minus three, zero, is outside that. Mm. So, it can't be, it can't be something you can achieve. Right, there's no way to make this minus three, zero, certainly not by choosing any of these points, but also no combi, no convex combination, no probabilistic combination of them, is going to do that either. Right. So, this one is just right out. Right, so that leaves minus four minus four, and it seems pretty. That's inside the convex hull, so there is some combination of them that would work. That's right. Any, any sense of what it would be? 2 3rds, I think 2 3rds D,D 1 3rd C,C. And I get that by noticing that D,D is minus six, minus six, and C,C is minus one minus one, and four minus four is two thirds of the way between there. Boom. Cool. Alright, so you're good with that? I'm good with that if you're good with that. It's your quiz. Yeah. I'm excited. Oh good. So those, it's this, this, the minus one minus one and the minus four minus four are, as as you pointed out there is a more general result here. Having to do with the convex hull. Right. Alright, so through the magic of computer graphics, I have a slightly better depiction of this particular region now. As you pointed out this, this convex hull of the points is really important and what it represents is the, we can call it the feasible region, these are average payoffs of some joint strategy they, they may have to collude to do this. And they may not be particularly happy to do that. [LAUGH]. But the fact of the matter is, they can achieve, by working together, payoffs anywhere inside this region. Huh, so that's what my student Liam always meant when he talked about feasible regions. Maybe so, I mean it could be that he meant any number of other things like places he's willing to live when he goes to get a job. No, it was all game theory stuff, I just never knew what he was talking about, but now I do Michael, thanks to you. Sure, hey I'm, I'm happy to help. So this is a really useful kind of geometric way of picturing something that would otherwise be a little bit harder to see, I think in the matrix form. Sure.

P. Minmax Profile Question

All right, the next concept that we're going to need to understand the folk theorem is the notion of a minmax profile. So a minmax profile is going to be a pair of payoffs, one for each player. And the value for player represents the payoffs that that player can achieve by defending itself from a malicious adversary. So what do you suppose a malicious adversary would mean in a game theory context? Someone who's desperately trying to hurt you. And what does hurt mean? Gives you the lowest score. Yeah, and what does that remind you of from your lesson? My grad students. You think they're malicious? It would explain a few things. Yeah, I don't think they're malicious. They're sweet. [LAUGH] Yeah, I know a lot of them and they're, they're wonderful people. Well, what it reminds me of is, they are wonderful people. It reminds me of zero-sum games. Exactly. So you can imagine thinking about the game that we're playing, now, no longer as being I get my payoff and you get yours, but I get my payoff and you get the negative of my payoff. So you don't, you don't really care about yourself anymore. All you care about is hurting me. And that's, that's the idea of a malicious adversary. I have some ex-girlfriends like that. I'm so. Oh. [LAUGH]. It is. People do get into this mode sometimes. And that, that's actually going to be important in understanding the folk there. Hmm. So what I'd like to do is figure out what the min-max profile is for this game. So this is a very famous Game theory, game example. Sometimes goes by the name battle of the sexes. That what the b and the s stand for? Sometimes it stands for Bach and Stravinsky. Blech. Those are like composers, I think. You mean like the Backstreet Boys and Sting. Ahh. Alright, that works for me. So, so, let me explain this story. It turns out that Smooth and Curly actually got away, they didn't make any kind of deal, they actually just figured out a way of escaping from the jail. So they're, they're back out on the streets again, and they decided that they'd like to celebrate their freedom by going out to see a concert. And they both decided that in advance, but what they didn't know was which concerts were available. Once they escaped out into the world, they couldn't communicate with each other, they discovered that there's in fact two concerts in the city that night. The Backstreet Boys are playing, and Sting is playing. Okay. Now as it turns out, each of them is now going to have to choose, whether to go to the concert with Backstreet Boys or Sting, and they're choosing independently. Now, if they end up going to different concert's they're both going to be unhappy and get zero's. I see. If they end up at

the same concert, then they're going to be happier, but in fact, as it turns out, Smooth really likes the Backstreet Boys and would prefer that they both end up at the Backstreet Boy concert. But Curly really likes Sting and would prefer that they end up at the Sting Concert. That's not realistic. Which part? The fact that I prefer the Backstreet Boys to Sting. What, what do you mean you? I mean this is Smooth. He's a criminal. Mm, that's a fine point that you make there. There is no connection between these characters and ourselves. Real life characters, living, dead or fictional, or mathematical, or instructional. If so, otherwise purely coincidental. Yeah. I could switch these around if you'd prefer. No, no. I'll go with your fantasy. [LAUGH] All right. I, yeah, I think that payoff matrix may look something like we both have twos in, in the s, the same place. Mhm. But anyway, but let's say for the purposes of this example, there's a little bit of a disagreement. Okay, so now what we need to figure out is what the minmax profile is for this game. Okay. Alright, so that's going to be a pair of numbers. Mm-hm. One number corresponds to the payoff for Curly and one number corresponds to the payoff for Smooth. And it should be, the payoff for Curly should be the payoff that Curly can guarantee himself even if Smooth is trying to get him to have a low score. Okay. And vice versa. Smooth's score is going to be the score that it can guarantee, he can guarantee himself even if Curly is trying to minimize Smooth's score. All right. So let's do this as a quiz. So,I want you to find the min-max profile for this game, this Bach, Stravinsky game or Backstreet, Sting game, and put the, the number for Curly in the first box and Smooth in the second box. Okay. Go.

Q. Minmax Profile Solution

Alright, what'd you got? I'm going to say it's one, one. Alright, how would you figure that out? Well first, I would ask you if that's correct. [LAUGH] [LAUGH] We'll see when you try to. [LAUGH] Figure it out. Okay, so, the idea here is I'm trying to figure out what Curly would get, if Smoove was out to get, make certain that Curly got the worst possible value. So, Curly gets to ch Curly gets to choose among rows. And Smoove gets to choose among columns. So, If I chose the B column, then Curly could get a one. You're given choice Curly would go for the first row instead of the second row, because Curly would get a one. If I took the S column, as Smoove, Smoove took the S column, then Curly would choose the second row, and would get two. And one is lower than two. So, Smoove would choose the first column and Curly would have to get the first row and would end up with a one. That make sense? Yes, but here's what you haven't figured out. Yes? What happens if Smoove, no because, because Smoove says, I'm always going to choose the Backstreet Boys. Mm-hm. Then you're right, Curly should also choose that and at least coordinate. Mm-hm. But what if Smoove is random say half and half random between the two options. And I don't know what's going to happen before then? And you don't know what's going to happen, sorry? I know that, I know it's half Curly knows it's half and half but doesn't know which one he's choosing any given time? Exactly. I see. Then, I would end up with, one half and one. Then for choosing B. Uh-huh. Expect the score would be a half, and for choosing S, they'd expect the wor, expect the score would be one. Yep. Oh, I see. And, and then Curly would choose S and get the one. Right. And that's still consistent with, with that? Mm-hm. Can we work out what the what the worst case is? What do you mean? Well I feel like, we should solve it like a [INAUDIBLE] game, like we did in your lesson. Oh. Say that Smoove is actually choosing a pro, probability either X or 1 minus X. So I thought you were asking for like a pure decision, I wasn't even thinking about mixed decisions. Uh-huh, yes, it could be a malicious and randomized adversary. [SOUND] So we are. [LAUGH] You said yuck [LAUGH]. We, we are really talking about my ex-girlfriends. Okay so you just do the math. Do the math. Alright, so if we, if Smoove chooses Backstreet Boys with probability X and Sting with probability of one minus X. Mm-hm. Then, Curly for choosing Backstreet Boys will get X on average and for choosing Sting, we'll get two times one minus six on average. And the useful point is going to be to discover when these are equal to each other. Yep. So in fact, Smoove, by being malicious and stochastic, can actually force things down to 2 3rds. Hm. And things being symmetric as they are, Curly can do the same. Okay. So, basically, Curly can behave in such a way, that even against a malicious adversary, it could, he could guarantee himself to a score of 2 3rds. Yeah, a malicious possibly mixed adversary. That's right. Okay. But one, one would be right if we were sticking with pure strategies, but why would we do that? That's right, but for the purpose, if that's right, and you can do a version of the folk theorem. In fact, there's lots of different flavors of the folk theorem. The one that we're going to focus on is going to allow for these mixed strategies. Mm-hm. But. In fact in general you could say you know, no I kind of like the mixed strategies, let's just stick with that. No, I like the mixed strategies too, I just wasn't thinking about them. So, I want to point out that in fact the solution that you gave, I think that actually does correspond to what is usually called Minmax, which is the pure strategy. Yeah! So the, [LAUGH] the minmax is, is in fact one, one. The other concept is really important too though. And I think it's sometimes called the security level profile. So instead of the min max level profile the security level profile. And that allows for the possibility of, of mixed strategies. So that gets you down to the 2 3rds, 2 3rds. I think you know, it turns out that there's folk theorems that can be defined with with either of those concepts. I prefer this one. But I do like this name better [LAUGH]. So I, I apologize if that made things confusing. I'm not confused now. I think in the end Michael, the important thing is we were both right. Well, the example that I'm going to next, these two concepts line up. So let's let's do that, and then we don't have to care. [LAUGH] I, I'm all for that. Let's do that.

R. Security Level Profile

So, here we are, back at the prisoner's dilemma, again. You may recall this picture. Vaguely. Let's add to this, the minmax, or security level profile. So, for prisoner's dilemma, what is the, the minmax? Isn't it d comma d? It is indeed. D comma d. Right so this is value that you can guarantee yourself against a malicious adversary. Malicious adversary is just going to defect on you, and the best thing you can do in that case is defect yourself. Yep. Agreed, agreed? Agreed. Alright, so now let's take a look at the intersection of the following two regions. There's this nice yellow region that we've already got, and then we've got a new region that's defined by this minmax point. This minmax profile. So the region that is above and to

the right of this, of this minmax point. Mm-hm. So, the, that's the region. This, this region, alright we already said that this yellow region is called the feasible region. Mm-hm, or orange or whatever color it is. So, I'm thinking we can call this other region the acceptable region. And it, and, the, what I mean by that is if you think about it, payoffs in this region are, smooth, getting more than what smooth can guarantee itself in an adversarial situation. And Curly getting more than Curly could guarantee himself in an adversarial situation. So, these are all, like, you know, better than it could be. So, why not call it the preferable region? The preferable regions, preferable to not being in this region. Mm-hm The intersection of these two Is the feasible preferable acceptable reason? [LAUGH] Exactly. It's kind of, you know, special, from the perspective that it is both feasible and preferable. And now we are ready to state the Folk Theorem.

S. Folksy Theorem

So here's the Folk Theorem. Any feasible payoff profile that strictly dominates the minmax or security level profile can be realized as a Nash equilibrium payoff profile, with a sufficiently large discount factor. Prove it. What we're going to do is we're going to construct a way of behaving where both players are going to play so that they achieve this feasible profile. And the reason to make that a Nash equilibrium, what we need to do is make it so that it's a best response. And the way that we are going to make it a best response is we're going to say do what you're told. Follow, follow your instructions to achieve that feasible payoff. And if you don't, then the other player is going to attack you, is going to adopt a strategy that forces you down to your minmax or security level, and that's your threat. So the best response to that threat is to just go along and, and and do what you're told to achieve that feasible payoff. The only way that that's going to be stable though is if the thing that you're asked to do, the feasible payoff, is better than the minmax, right? Because that has to be a threat. You can't threaten somebody and say, you know, do this or I'm going to give you candy. It's gotta be do this or I'm going to do, give you something that's less pleasant than what I've asked you to do. Okay, that actually makes sense. Yeah, so this is, this is a really cool idea. I like it. Hey, could you try saying that again, but with a Southern accent, just the Folk Theorem part? Any feasible payoff profile that strictly dominates the minmax/security level profile can be realized as a Nash equilibrium payoff profile with sufficiently large discount factor. I like that because now it's a folksey theorem. [LAUGH]

T. Grim Trigger

So another way to think about the proof of the folk theorem, is you could prove it with a little strategy that is referred to as grim trigger. Mm. I like that. I like the way it sounds. So here's, here's the basic structure of grim trigger. It says that what we're going to do, is we're going to start off, taking some kind of action or pattern of actions, it's a mutual benefit. And as long as it's cooperation continues this mutual beneficial, behavior will continue. But however, if you ever cross the line, and fail to cooperate with me. Then I will deal out vengeance against you forever. Hm, so once again, we're talking about my ex girlfriends. I don't know why you're obsessed with this. [LAUGH] Maybe it's just, maybe it's just trying to help you understand. So you know kind of what this situation is. Anyways here's, here's what that looks like in the context of prisoners dilemma. Alright so cooperation is the, is the mutually beneficial action. Yes. And as long as you continue to cooperate with me, that's this C arrow here, then I will continue to cooperate with you, but if you ever defect on me, I swear I will spend the rest of my life making you pay. So no matter what you do at this point, defect of cooperate, I will just continue to defect on you. Pain will rain from the sky. Okay, well this makes sense. So, the idea here is that if you know that I"m going to do this, then hopefully it makes sense for us to continue to mutually benefit. Right. So the whole purpose of this is to create, a, a Nash Equilibrium System kind of situation. Right? Where if I'm playing this strategy. And you're playing this strategy, then neither of us has any incentive to cross the line, and so we're just going to continue to cooperate. Crossing the line is going to decrease your reward, so there's no benefit to doing it, so you won't do it. So it, it's nice because it gets us a Nash Equilibrium. Hm. But there's a problem with it. Of course. And you pointed it out before, so let's, let's dive in and make sure that we understand it and see if we can fix it. Okay.

U. Implausible Threats

The problem is that in some sense, the threat is implausible. And it, and it, in a very kind of real sense. So what's happened is that if you do fake out on me, if you do cross that line, the idea that I will then spend the rest of my days punishing you, forgoing reward myself, right? Not taking the best response against you, seems kind of crazy. Do you agree? Yeah, again, my ex-girlfriend. Yeah, I totally get this. No, but no, I'm saying that nobody would do that. Right, so it, it would be like being in a elevator with a stick of dynamite and seeing that someone has a hundred dollars and saying, give me your hundred dollars or I'll blow us both up. That's not really a reasonable threat because the alternative to me not giving you a hundred dollars is, you die, which seems probably not worth it. That's right, so you could think about the possibility of, okay, I'm going to not give you the $100, you say that you're going to blow me up, but you will hurt yourself more than you hurt me, so it won't be a best response. Not blowing me up and just not getting the $100 and leaving the elevator is better for you than blowing me up. Right. So that is an implausible threat. So the way we formalize this idea, in the game theoretic context, is to say that we're interested. A plausible threat corresponds to something that's called a subgame perfect equilibrium. Okay. So subgame perfect means that each player is always taking a best response, independent of the history. All right, so let's actually look at a concrete example here, let's imagine playing Grim Trigger, against Tit for Tat. So my first question to you is, are these two strategies in nash equilibrio with each other. Yeah I guess so. And why is that? Because, the fact, if I'm playing Tit, if one player is playing Tit for Tat then the Grim Trigger thing doesn't matter anyway because both of you are going to cooperate forever and it doesn't make any sense to deviate. Right, so any strategy that I could choose that's different than Grim Trigger is going to on average do no better, possibly worse. Right. So I might as well stick with Grim Trigger and Tit for Tat has the

same kind of feeling about it, that, it's cooperating with Grim. And any, it can't really do anything better so it might as well do that. Right. So the next question to ask is. Are these two strategies in a subgame perfect equilibrium with each other? And the way that you actually test that, is you say, well, they are not subgame perfect. If there's some history of actions that we could feed to these machines, so that, so that, you know, here's, here's what Grim is doing. It's, it's some, some sequence of cooperates and defects, and here's what Tit for Tat is doing. It's some sequence of cooperates and defects. And once we've reached some particular point. Is it the case that one or the other of these machines is not taking a best response that it could actually change it's behavior away from, what the machine says and do better then what the machine says. If that is the case then it's not subgame perfect but if it's the case of all histories. They're always taking a best response. Then, it is subgame perfect. So, so do you see a, a sequence of, of moves that these two players can take where one or the other of them is not going to be doing a best response? It's, can take, right? As opposed to, will take. I don't understand. Yeah, I'm not sure I do either. That's why I asked the question. It's not a, you know made up history, it's like an actual set of moves that are consistent with Grim and Tit for Tat. No no, no no, so it is, it is not necessarily. So we know that if we actually play these against each other the only history that we're going to see, is. Cooperate forever. Right, Grim is going to do cooperate cooperate cooperate cooperate, Tit for Tat is going to do cooperate cooperate cooperate cooperate. And so they are, and everything's fine. The question is, can we actually go in and alter, the history, so that one or the other in the machines could take a better action than the one that the machine tells it to take. Yeah if Tit for Tat, ever does defect. Alright, so let's take a look at that. So, let's say, on the first move Grim cooperates and Tit for Tat defects. Okay, so let's say that, that's the moment in time. What will the machines do at this point? Well, at this point the and next time step, Tit for Tat will cooperate and Grim will defect. Good and then thereafter. Grim will always defect. And then Tit for Tat will always defect. Right. So the pay of that Grim gets at this point is going to be, well initially high but then very very low. Mm-hm. On the other hand could Grim have changed it's behavior to do better than this? Yeah. Just by doing just, by choosing to cooperate. By choosing to cooperate, so it sort of ignore the fact that, that Tit for Tat did the defect, and instead do a cooperate here, then Grim would do better. So the idea is that Grim is making a threat, but when it comes time to actually follow through on that threat, it's actually doing something that is worse for itself. Than what it would do otherwise. Do, do you see that? I do. So is it subgame perfect? No. And the proof of that is exactly, exactly what you said, Take, take a look at this history. Here's a history where Grim would not be willing to actually follow through on its threat. Right. So it's an implausible threat, and that's bad. So maybe we've now just undone all the awesomeness that we had done. No. Well maybe. I mean the awesomeness was hey look we can actually get machines that are in nash equilibrium and they're cooperative in, in prisoner's dilemma so they're actually kind of doing the right thing. And, turns out well they are but they're, depending on this notion of implausible threats to do it. Mm, how should I feel about that? Well, let's see if we can fix it. Okay.

V. TfT vs. TfT Question

All right. So let's make sure that we get this concept. So let's evaluate, tit-for-tat versus tit-for-tat, spy versus spy, and ask whether or not they are subgame perfect, or in a subgame perfect equilibrium with each other. And your choices are yes and no. Mm hm. But if you say no, I'd like you to give me a sequence to show that it is not subgame perfect. In other words, that if they were to take this sequence of actions and this one was to take this sequence of actions, it would leave the machines in a position where they would not be willing to follow through on their threat. That it would be better for them to do something else in the long run, assuming that they're still playing against the other tit-for-tat machine. From that point on. Yup. Just imagine that someone could go in and change one thing. Would you still want to follow tit-for-tat the next time step or not? Yeah, I don't know if that has to be just one thing, but yeah. That's right. Change, change the sequence leading up to this point and then say, okay, do you still want to do what tit-for-tat is telling you to do, or would you rather do something else? Right. Okay. I think I got it. Cool. All right. Go.

W. TfT vs. TfT Solution

Okay. What's your answer? My answer is no. No, I'm sorry, that's wrong. No, no, I think that, well, if it's, if it's right, then we need to provide a sequence that proves it. Okay. So what are you thinking? Well, I was thinking actually something very similar to what we just saw. Okay. Where so tit for tat, what they're going to do is they're going to do cooperate, cooperate, cooperate, cooperate, right? That's what they normally want to do. Exactly. So what would happen if at one point, one of them defected? Okay, just for simplicity let's make it the very first point. Mm-hm. So, tit for tat number one defects, and tit for tat number two also defects. At the very first time? No, it cooperates, because it's done at the same time. Well, I mean so we're, we can feed it anything we want. So we could tell tit for tat two to cooperate. So it's sort of like we've taken over its brain for a brief amount of time. Right. So I'm not yet convinced it's going to matter for this, but the thing is that from that point on, tit for tat two is going to want to, for the next step tit for tat two is going to want to defect. That's right. And, tit for tat one would want to cooperate. Uh-huh. And that's sort of. Sucks. [LAUGH] For tit-for-tat two, right? So, I don't know, so maybe we should try to think through. What is the expected reward, for tit-for-tat two, to actually do this defect at this time? Or wait, no. So, so, sorry to, to stick with the tit-for-tat machine at this time. Well, what's going to happen at that point is, it's going to keep alternating. That's exactly right. So it's going to get the, the rewards corresponding to D versus C, C versus D, D versus C, C versus D. Over and over again. Yeah. So let's thi, let's think about it in the average reward case. So, in the D versus C, if it does D when the other machine does C, then it gets zero. Mm-hm. If it does C when the other one does D it gets minus nine. Mm-hm. And then this alternates. So if we look at the average award, which is basically what you get when the discount factor's very, very high. Mm-hm. It's scoring negative 4.5. Right. Is there any way, that it could behave against tit for tat, starting from this point that would do better than negative 4.5. Just go ahead and cooperate. Just cooperate forever? Well cooperate the next time and then keep doing tit for tat from that point on. It'll work out to be cooperate forever. On average.

That's right. What will get is a minus, minus one. So not being tit for tat at that point but instead, instead turning always to cooperate would actually get it better. So the idea that is should defect at this point, is an implausible threat. Exactly. So this is not sub-game perfect. So yes, you nailed it. Yeah! Does that make sense? It did make sense. Good, alright. So that leaves open the question of, is there a way, to be sub game perfect in Prisoner's dilemma? Can I ask you a question? Sure. Before you answer that. So, I had sort of convinced myself that it didn't matter whether tit for tat number two started out with C or started out with D. I'm trying to decide whether that's actually true. 'Kay, that's a good question. So what will happen, at this from this point on if we now continue. We, you know, we took over the brains for tit for tat, and we forced them to play defect against defect. Mm-hm. And now we release that, and we let them do what, whatever it is that they're going to do. And what is th, what are they going to do? Is [CROSSTALK] They would defect forever [CROSSTALK] Defect forever. Yeah. And is there anything that tit-for-tat two machine, could do to get a better score than that? Cooperate. Yeah, so it could. Cooperating with tit for tat will bring it back into mutual cooperation. Hm. It will actually get a better score. Yeah. So, in one case, it would average to minus one and the other one, it would, it would average to minus three and minus one is better, so, you're right. Good point. Okay. So what matters is that we get we get them defecting. Right. Okay, so that makes sense. So, I, I was right that it didn't matter, although you do get slightly different answers, or slightly different averages. That's right. But in both cases, there's a way of getting a higher average. Right. Okay, cool, that's what I thought. I thought it was something like that. So now, let's go back to what you wanted to do. So, are we going to be able to figure out how to do Prisoner's dilemma in a way that is sub game perfect? Well, how about I propose a machine, and we'll see what it does? Okay.

X. Pavlov

So here's a machine that is sometimes referred to as Pavlov named after the Russian psychologists who was studying gastric juices and then figured out how animals learn. So I don't know why it's called that in this particular case but, here's what the machine looks like. It says start off cooperating and as long as the opponent keeps cooperating, then cooperate. So, so far it sounds a lot like tit for tat. Yeah. If you defect the then move to the defect state. So, okay still looks like tit for tat. And again, this defect state has two arrows coming out of it. But their reversed from what they were in tit for tat. Here it says, if you cooperate with me, I will continue to defect [LAUGH] against you. But if you defect against me, then I'm willing to cooperate with you. So, that's a little strange, right? That is a little strange, yes. So, you know, the way to get me to stop defecting against you is to defect against me. And then, I I become cooperative again. So, in other words, take advantage of you until you sort of, pull a trigger on me. Seems like it. Yeah. It's a funny thing. So so again, so tit for tat is like repeat what the opponent did. Pavlov is basically cooperate as long as we're agreeing. If we both defect, I'll cooperate. If we both cooperate, I'll cooperate. But if we disagree, if I dis, if I defect and you cooperate or you cooperate and I defect, then I will defect against you on the next round. That sort of makes sense. Really? Yeah, right? I mean, it says: if you're cooperating, I'm going to cooperate. If, we're both going to start out cooperating. And then if you ever defect on me, then you have basically attacked me, and so I'm going to defect on you. Unless you start cooperating again, in which case I think that you're, you're being reasonable, because of what I did, and so now we're going to start cooperating again. Except that's tit for tat [LAUGH]. No, you're right. You're right, I, I take it back. It doesn't make any sense, too many. Yeah, it's weird. It's a little bit weird. too many V's. Too many V's. Too many V's? Mm-hm. In Pavlov. Yes. Two V's. Yeah, but you can think of them being like arrowheads. Oh well then that makes much more sense. Yeah, exactly. Okay. Yeah, so it's this weird thing where I'm going to continue to defect against you until you realize I'm hurting you, you punch me once. [SOUND] And now okay, good, we're even again. We good? Yeah we're good. Oh this is like men on a basketball court. Yeah, sure but, now here's the question. Is this Nash? So, we'll make that a quiz.

Y. Pavlov vs. Pavlov Question

So, here's a quiz. And this is, I chose it this way so that I can maximize the number of v's in one sentence. Is Pavlov v. Pavlov, Nash? [LAUGH] I think I know the answer. Alright. Let's, let's give everybody else a chance to think about it. Okay. Go.

Z. Pavlov vs. Pavlov Solution

Alright, what'd you get? I say yes. And the answer, that comes from what? Well, we both start off and cooperate. And so, you are always going to cooperate and it doesn't make any sense for anyone to ever move away from cooperation. Indeed. Hence, you are at equilibrium, in particular, you are at Nash equilibrium. That is correct.

AA. Pavlov Is Subgame Perfect

So, that's very good. So, you, were able to realize that Pavlov is in Nash equilibrium. But we can go further than that. Or farther, than that. Further. We can go further than that. And show that in fact Pavlov is sub-game perfect. So, unlike tit for tat. It's actually subgame perfect. So let's let's take a look at how we could convince ourselves of that. So I think the important thing to see is that by feeding these two Pavlov machines different sequences we can get into any of these four different combinations of states. They're both in the cooperation state. They're both in the. Both in the defect state, ones cooperate, ones defect, ones defect and ones cooperate. Mh-hm. so is it the case, that no matter which state that those are in, that the average reward is going to be mutual cooperation. So let's check, if they are both cooperating, and we continue with these Pavlov machines then they will mutually cooperate, so yes. Mh-hm. Alright if they're in defect defect then what's going to happen? Well then they both agree, so then they cooperate. So they're going to move to cooperate and then they'll stay there

forever, so then that'll be mutual cooperation. Awesome. What if ones in cooperation and ones in defect? Then they disagree. Right. And they move to the other state. More specifically what? Oh, I don't know [LAUGH], I can't remember. I'm trying to keep track of who, who's who. So if I cooperate. and you defect, then let's see the guy who cooperates moves to defect. And the guy who defects moves to defect. Because you, and now you agree, and so you're going to cooperate. Boom. So, when we're in the cooperate defect state, then on the first move. Let's see, you just, yeah, the right-hand Pavlov just defected, so that causes this transition. And this guy just cooperated, which causes this transition, so that we've gone to defect defect. Right. Which means that we're going to average cooperation, because [CROSSTALK] Mm-hm. That's where we're going to get stuck in the long run. Right. And the same thing works through here. Boom. That's actually very cool, and kind of counter-intuitive. Yeah. And truly neat. So sort of no matter what weird sequence we've been fed we manage to resynchronize and then return to a mutually cooperative state. So I have a question for you. Go for it. So presumably this is really cool like mathmatically because now we shold all do, we should all be Pavlovians. Like we're all Kinseans. ANd then we just kind of move forward from there. D people do this? I don't know the answer to that question. I mean other than men on a basketball course. Sure, you can always return to that. Though I'm not sure I'm aware of any analyses of men on a basketball court and whether or not You know, people have analyzed that. Hmm. But how about this. If I find out, I will post something on the instructor's comments. Okay, that sounds reasonable. So Pavlov is subgame perfect. That's awesome. So remind me again why I care that something is subgame perfect? Because it means that, so let's say that you actually, so I'm being this left Pavlov and you defect and me and you're like yeah I'm going to defect on you because I just want to take advantage of you, and you're, you're going to forgive me and I will have gotten this extra bonus for that. And what it turns out is that No, if we do Pavlov versus Pavlov, we're going to fall into mutual cooperation no matter what. So, so, so this defection that I do, this, this threat, this punishment that I deal out to you, you can earn it back, and we can go back into a cooperative state. Right. So, it's worth it to me to punish you, because I know that it's not going to cost me anything in the long run, and it stabilizes your behavior in the short run. Sure. It makes perfect sense. So it becomes a plausible threat. I like it.

AB. Computational Folk Theorem

So this Pavlov idea actually is more general than just the prisoner's dilemma or iterated prisoner's dilemma. And in fact, led to a result that I like to call the computational folk theorem. The idea of the computational folk theorem says that you give me any two player, bimatrix game. What's a bimatrix game? Just that there's two players. [LAUGH] Okay. [LAUGH] So it seems kind of redundant, doesn't it? It does. What makes it bimatrix as opposed to two player zero sum game which you can write down with a single matrix, this is like, each, each player has its own reward matrix. I see. But you're right, I should have, I could've just said bimatrix game and left out the two player. And it's an average reward repeated game. So we're going to play. Rip, Round after round after round. And we're going to look at the average reward. Or, you can also think of it as discounted with an extremely high discount factor. Okay. So you give me one of those games. And what I can do is, I can build a Pavlov-like machine for, the, for any of these games. And use that to construct a subgame-perfect Nash equilibrium, for any of these games, in polynomial time. Wow. And, so the way that this works is if it is possible for us to have some kind of mutually beneficial relationship, then I can build a Pavlov-like machine. Quickly. If not, the game is actually zero sum like, right? because in a zero sum game we can't mutually benefit, so we cant do anything like Pavlov we're just going to beat each other up. So we can actually solve, linear program in polynomial time, and work out what the strategies would be if we we're playing a zero sum like game. And so either that works, and produces a Nash equilibrium, and we can test that. Or it doesn't work, but at most one player can improve its behavior. And by taking that best response against what the other player does in a zero-sum like sense, then that will be a Nash equilibrium. So there's three possible forms of the Nash equilibrium. But, we can tick through these, figure out which one is right and drive the actual strategies in polynomial time. Wow, that's pretty impressive, who came up with this idea? So this is a result due to Peter Stone and somebody, Oh yeah me. Oh, well that's very impressive, so you managed to find a way to sneak in some of your own work into this class? Here, let's do some more of that. Okay, I'm a big fan. And I think that's fair because I did that way back when on mimic. Mimic. So, yeah, so what the last topic that, this is, that's all I really wanted to say about the Folk theorem and repeated games. What I'd like to do now is move to stochastic games, which is a generalization of repeating games. And, talk a little bit about how this relates Back to things like queue learning and MDPs. Oh, okay. That sounds cool, almost sounds like you're wrapping up. It is, that. And that will be the end of, end of the new material. Wow. Well that means we're coming towards the end of the entire course. I know. We're going to all cry with, disappointment. And I think. And, and I just say this, you know, as a, as an idle suggestion, that the students should demand that we teach more classes. I concur. So let's get there so that they can demand. [LAUGH]

AC. Stochastic Games and Multiagent RL

So what I would like to tell you about is a generalization of both MDPs and repeated games, that is, that goes by the name of Stochastic games, also sometimes Markov games. Mm. I like the name Marcov game better, but I used Stochastic game because that's what people call it and sometimes it's good to use words that other people use. And what what Stochastic games give us is a formal model. For multiagent reinforcement learning. In fact, I like to think of this in terms of an analogy. Which is something like MDP is to RL as stochastic game is to multiagent RL. It's a formal model. That let's us express the sorts of problems that take place in this formalized problem setting. Hm. That sounds very promising. Cool. Alright so let me let me give you a, I'll start off by explaining it in terms of an example and then I'll give a more formal definition because you know, I can't not. So so this is a little game played between A and B. Oh, I should have it between smooth and curly, but At the traditionally it's played between A and B. Mm, and sometimes it's good to use the words that other people use. [LAUGH] I've heard that. I wouldn't say it quite that way. So this is a three by three grid each of the players can go north,

south, east and west. And can stay put if that's helpful. And the, the transitions are deterministic, except for through these, these walls here which are called semi-walls. Mm-hm. So these thick lines represent walls that you can't go through, the thin, wall, lines just represent cell boundaries, but this kind of dashed line here is a semi-wall, and that means If you try to go through that, say by going north from, if A goes north from this position, then 50% probability A will actually go to the next state, and 50% probability A will stay where A is. So, the goal is to get to the dollar sign. And if you get to the dollar sign you get a hundred dollars. So if we ignore A for a second, what should B do to minimize the number of steps necessary to get the reward. Go left, and then go up and go up. Oh, I'm sorry. Go west, and then go north and then go north. Yeah, and what should A do ignoring B? Go east and then go north and then north. Yeah. Unfortunately these guys live in the world together, and what happens is, they can't occupy the same square. And as soon as somebody reaches the dollar sign the game ends and the other player, if the other player hasn't reached the dollar sign, gets nothing. I see. So now there's a little bit of contention. So what happens if A and B both try to go, to the same square at the same time? Let's say that we flip a coin and one of them gets to go, first and then the other one will bounce off of the first one. But that's not a problem when it comes to reaching the money. But it's not a problem, yes, right, so the money is kind of like a money pit. [LAUGH] I don't think that's what a money pit is, but okay. And so they can dive in and they both get the money, because they're in the money pit. I like it. So what do you do if you're A? How do you play this game? Oh! Let's think of another thing. Is, can you think of what, what it might mean to, to have a Nash Equilibrium in a game like this? Oh, that's an interesting question. It would mean, well, it would mean, well, what do you mean, what would it mean? It would mean that, neither one of them would want to deviate. It would mean a pair of strategies for the two players. Now the strategies are now multi-step things that say, they're like policies, right? So... Yeah. Like it's a pair of policies, such that neither would prefer to switch. So can you think of a pair of policies that would have that property. Well, no I'm not sure. I was trying to think about that. I was thinking that kind of, if I were a nice guy what I would want to do is I would want us both to try to go through the, the semi walls, and if we both go through the semi-walls we just go up again and then we, we hit the dollar sign at the same time. And that's very nice. So okay, good. So that, that seems like a cooperative kind of strategy, right? Where they're both you know, 50% oh I'm sorry, 25% of the time both will get through, both will go to the goal together. Hooray. But... 25% of the time neither one will get through and then we're in the same place we were before, so that's okay. That's right. The problem is the other 50% where one of them gets through and the other one doesn't. Right, so what do, what you do if you make it through and the other one doesn't? What do I do, if I get through, and the other one doesn't? Well if I am only going to do this the one time then I just keep going and get the dollar, and the other person loses. Yeah, alright, so what this works out to be, is that A is going to get to the goal. 2 3rd's of the time, and B is going to get to the goal 2 3rd's of the time. Mm-hm. So, alright, so if that's the case, if I say, okay, A, that's what you should do, B, that's what you should do. Then is there a way that either A or B can switch strategies and do better? Well, if B, for example, decides to go west and then go up, what happens? Yes, that's a good question. B will now make it to the goal a 100% of the time, and A will only make it to the goal 50% of the time. So B has an incentive to switch to that, to this strategy if we tell them to both go through the semi-wall. Right. So that wasn't a Nash Equilibrium. B would wanted switch this new policy. Mm-hm. Is this a Nash Equilibrium? No. Wait, is it? No. Because, why doesn't A just choose to go west east? Well, would, would A do better on average by switching to this strategy? Well let's see. no, actually. Oh, no, no, you said half the time they go through. Yeah. So half the time you flip a coin. So half the time I don't make it. Right. But half the time I do. Right. So, actually, it looks the same. It looks the same. That's right. And B would go from 1 to one half. Yeah, that's true. [LAUGH] So, it, A doesn't have an incentive to do it, but B is hoping very much that A doesn't do that. Right. So so, yeah. So that, so there's one Nash Equilibrium where B takes the center. Another one where A takes the center. I guess if, if they do, if we do this coin flip thing, it, it works out this way. If it's the case that if they both if we change the rules here. So that if they collide, neither of them gets to go. Then go, both trying to go to the center is not a Nash equilibrium anymore, because you can do better by actually going up the semi-wall. Right. And so if we, if, if collision means nobody goes through, then, suddenly, you'd want to do the other thing. Exactly. Or one of you goes through the semi-wall and one goes the direct way. Right. So we can see that there's a bunch of different Nash equilibrium here, sorry, Nash equilibria here. And that it's not so obvious how you'd find them, but it is at least clear that they exist and they have a different form than what we had before, because they're not policies instead of these otherwise simplified just you know, choose this row of the matrix. Mm-hm. Cool. Alright. So let's think about how we might learn in these kinds of environments. Oh, okay, I like that already.

AD. Stochastic Games

So, stochastic games, originally due to Shappley, have a bunch of different quantities. State, actions, transitions, rewards, and discount factors. And here's how we're going to do it. We're going to, we're going to say that s, little s, is, is one of the states. And actions could be like little a, but actually, since we're going to focus mostly on two player games for the moment, I'm going to write actions as a and b. Where a is an action for player one and b is an action for player two. Sound okay? Sure. Alright. So next we have the transition function. So, the transition function says: If you're in some state, s, and there's some joint action that's taken, like all the players choose their action simultaneously, a comma b, then what's the probability of reaching some next state s prime? And we can write rewards the same way. So there's reward for player one, given that we're in state s and there's a joint action ab. And there's the reward for the other player, the second player. And a discount factor is you know, like a discount factor. Totally makes sense. So oh its the same discount factor for everyone. Yes! Good a good point. One need not define things that way, but in fact that is the way it's always defined. Hey, not to go of on a tangent here, but sometimes I see NDP's and things like stochastic games defined with a discount Factor being a part of the definition, and sometimes not. Like it's just a part of the prob, definition of the problem or sometimes its a parameter of an algorithm. Which do you prefer? Why do you, why haven't, why have the discount factor actually listed as part of the definition of the game? I have no justification other than it's nice to have listed the things that might be important as oppose

to you know, working through algorithms for while and then saying, oh yeah there's this other number that kind of matters too. Okay that's fair. I was just curious. So one of the things that I actually find really interesting is that this model was laid out by Shapley. Mm-hm. One of the, like a former Nobel prize winner. I guess once you're a Nobel prize winner, you're a Nobel prize winner forever. Yeah, [INAUDIBLE]. So, Shapley, the Nobel prize winner, and as we're going to see in a moment, this model is actually a generalization of MDPs, but Shapley published it before Bellman published about MDPs. Oh. This is pre-Bellman. So MDPs, to some extent, can be thought of as a narrowing of the definition of a stochastic game. Huh. So, all right. Let's do a little quiz. And see that we really understand the relationship between this model and other things that we've talked about. Okay.

AE. Models and Stochastic Games Question

Alright so, Stochastic Games are more general than other models that we've talked about. And so, just to make that case here's a way of making the Stochastic Game settings more constrained. And, by making them more constrained, actually turning them into other models that we've talked about or could talk about. So, I wrote down three different ways of constraining the Stochastic Game model. One says that we're going to make the reward function for one player the opposite of the reward function for the other. The next one says that the transition function has the property that for any state and joint action and next state. If that's going to be equal to the transition probability for state joint action, next state, where we've changed potentially the choice of action for player two. Mm-hm. So basically, player two doesn't matter to the transitions or the rewards for player one, and the rewards for player tow are always zero. So that's, that's again, you can specify this as a Stochastic Game and then in the third case we are saying that the number of states in the environment is exactly one. So I claim that by doing these restrictions. We get out the mark-off decision process model, a zero sum statistic game model, and the repeated game model that we've been talking about in the context of, like, the folk theorem. So, what I'd like you to do is write the letters, A, B, and C in the correct boxes. Okay. Go.

AF. Models and Stochastic Games Solution

Alright, talk me through it. Okay, so I'm going to say that I think I know the answers for this one. And let's start with the first one. So R1 equals minus R2, which you'll notice they're equal and opposite. And in fact if you add them up, that is you sum them you end up with zero. So I'm going to say that's a zero sum stochastic game. Nice. For two, basically you're saying that for all intends and purposes, there's only one agent. Which just makes it a regular Markov decision process. Yeah. So isn't that interesting? That just by the other player irrelevant, then that's what an MDP is. It's like a game where the other players are irrelevant. Yeah, which, both of my children are like that. But okay, I think that's pretty cool. And in fact, I'd be right in saying that R2 doesn't have to equal to zero. As long as it just equals to some constant. Yeah, that's, I mean, constant. Actually, depending on how you think about it, it could be, we could just ignore the whole R2 thing and just say that. As far as the first player is concerned, since the second player really has no impact on anything. It doesn't matter. But the reason I put that in is I got kind of scared that like. I feel like if I lived my life and knew that my actions effected the state and my rewards, but they were also effecting the rewards of somebody who didn't matter. Like I feel like that would actually still have an influence on me. Sure, but then the way you get around that is you would say, well, your R1 is actually equal to your R2. Oh. [CROSSTALK] It would somehow [UNKNOWN]. So, so if I had gone like that, wouldn't that be the case then that we're saying? Oh yeah, I see. That the second player is irrelevant, but the reward, but the first player may be relevant to both. Right. Yeah, okay, yeah I like that a little bit better. Yeah, I mean, once again, it all boils down to changing the rewards. Okay, and so given A and B, I know the answer to three must be C, unless you're tricking me, and it could be A or B again. And which I suppose you could have done, you didn't say they were mutually exclusive. So let me actually argue why it would be C? Well there is only one state and since you're in a stochastic game and you're going to be continually doing this. It means that you're basically doing the same game over and over and over again, so it's a repeated game. Yeah, yeah, yeah so in particular the actions impact the rewards, but they're not going to impact the transitions because you're always just going to back to where you were. The discount factor plays the role of, of decided when the game's going to end, stochastically. And, so yeah, it's exactly a repeated game, this is the one I feel most comfortable about because this really does recover that same model we've been talking about. I like it. Cool! Now, given that we actually are now in a model that generalizes MDPs, it would be nice if we can generalize some of the things that we did with MDPs, like Q learning and value iteration to this, this more general setting. So, that's what we're going to try next. Cool.

AG. Zero-Sum Stochastic Games 1

Now what makes stochastic games more interesting, perhaps, than repeated games, is the idea that the actions that the players take impact not just the rewards, but also future states. Right? And, so this is the same issue that comes up when we're talking about mark off decision processes and the way we dealt with it in that setting was by defining a value function. So, it seems pretty reasonable to try to go after that same idea again. So what I've got here is actually the Belmont Equation And let's look at this together and let's see if we can fix it because it's not quite right. Okay. For dealing with the idea of zero sum in a stochastic game. Okay so you remember the Belmen equation? We've got Q I star. Mm-hmm. So there was no I before, but Q star is the state. Is to find over state actions, so here we're going to define it over action, joint actions Mm-hm. For the two players, action pairs. The immediate reward to player i for take, for that joint action in that state plus the discounted expected value of the next state. So we need to factor in the transition probabilities So the transition of actually going to some next state s prime is s, sorry t of s, AB as prime. Right? So now, we're imagining what happens when we land in S prime. So what I've written here says, well, we're going to basically look a the Q values in the state that we landed in,

and kind of summarize them, summarize that value back up so that we can use it to define the, the value of the state that we left. You with me? I am with you. Alright so if we put this in if, if we say the way we're going to summarize the value for the new state that we land in. Is we think of it as actually a matrix game. That there's payoffs for each action choices of A prime and B prime. And over all of those, we need to kind of summarize well which of those actions in this table of values that we get for s prime. Which of those values are we going to propagate forward and call the value of that state? So what we did in regular MDPs is that we said we'll take a max over all the actions or in this case all the joint actions. Uhun. So what do you think that translates to? Well you wrote down max but that doesn't make sense, that doesn't, that can't be right. Well it translates, it means something. It just doesn't mean what we mean it to mean. That's true, that's fair. So what does it mean and then, and then, how can we fix it? So let's start off with what does it mean. It means that you kind of always assume that the joint actions that are going to be taken Will benefit you the most. So, everyone, everyone is trying to make you happy so, this makes you optimistic? yeah, sort of optimistic to the point of, of Delusion? Yes, very good. Right, it just basically says that whenever we're in a state, the whole world is going to choose their actions to benefit me, and this is not what we get a say a zero sum stacastic game, but a zero sum stacastic game, we should be you know. Like fighting it out at this point. So that would work out if everybody's rewards were the same, or everybody's rewards were the sum of everyone's rewards, or something like that. That's right. If it was some kind of team based game. Mm. Or if everybody was, you know, going to sacrifice their own happiness for the benefit of Qi, or i I mean. Hm. So it's not reasonable to assume that. In fact, what was it that we were assuming when we had a zero sum game that was just a single stage? Right? Just a single game and then we were done. Oh, that people were doing minimax. Right. And maximin. So what if we changed the equation to look like that? So, what I mean by this, is when, when we we evaluate the value of a state. We actually solve the zero sum game in the Q values and take that value and use it in the, in the context of this equation. That seems closer to right. Yeah. I mean, it's not an unreasonable thing to do. It's just to say, I'm going to summarize the future by imagining that we're going to play that game that represents, you know, all the future. Sure. And I'm going to act in such a way to, to try to Maximize that assuming you're trying to minimize it, which makes perfect sense if it's a zero-sum game. Right. I was, yeah, and we're still, we're still acting as if there are only two people here. Yeah, yeah, that's right. It turns out that when you're talking about zero-sum it really implies that there's only two players. Because if you have a zero-sum three player game, it really is just a general sum game. You can imagine that the third player is just an extra factor that's just messing with the numbers to make things sum up to zero. So, yeah, so zero sum really does kind of focus on this two player setting. That makes sense. So we got this modified kelvin equation and we can even translate it into a form that's like Q learning. So the analog of the kelvin equation and the Q learning update in this setting would be that we. If we're in some state, there's some joint action that's taken, there's some pair of rewards that comes and some next state that's visited that the Q value for that state, joint action pair, is going to be updated to be closer to, the reward for player i plus the discounted expected value, or sorry, the discounted Summarized value or v value of the new states as prime, and we'll again, we'll use mini-max to summarize what the values are in that new state. I like it. And that equation is sometimes referred to as mini-max Q, because it's like the Q learning update but just with the mini-max operator instead of a max. That makes sense.

AH. Zero-Sum Stochastic Games 2

So, if we set things up this way, we actually get some wonderful properties coming out. So here are some things we know about this set up for zero-sum stochastic games. Value iteration works, so we can actually solve this system of equations by using the value iteration trick, which is to say, we initialize these Q values to whatever and then we just iterate this as an assignment, right, we just say, you know, equals. So value iteration works. This minimax Q algorithm converges under the same kinds of conditions that Q learning converges, so we get this nice, you know, Q learning analogue in this multi-agent setting. The Q star that's defined by these equations is unique, so we iterate it and we find it and it's just, there's just that one answer. The policies for the two players can be computed independently, that is to say, if two different players are running minimax Q on their own and not really coordinating with each other except for by playing the game, that the policies that they get out will actually converge to minimax optimal policies. So it really does solve the zero sum game, which is maybe not so surprising because, you know, they are trying to kill each other after all. [LAUGH] Yeah. So, the idea that they'd have to collaborate to do that efficiently would be weird. [LAUGH] I never thought about it like that, but, yeah, that would be weird. The, this update that I've written here can be computed efficiently, which is to say in polynomial time. Because this minimax can be computed using linear programming. Yes of course. And, finally, if we actually iterate this Q equation and, and it's converging to Q star, knowing Q star is enough to figure out how to behave optimally. So we can convert these Q values into an actual behavior, again, by using the, the solution in the linear program. So it's just like MDPs of value iteration with Q-learning? Exactly. It's like we've gone to, to a second agent and it really hasn't impacted things negatively at all. This is, this is all the, pretty much all the things that we want, come out. There, there are some things that don't come out. For example, in the case of an MDP, we can solve these, this system of linear equations in polynomial time. Not just by value iteration, but we can actually set up it as a single linear program and solve it and be done in linear time or, sorry, not linear time, polynomial time. This is not known to be true in the zero-sum stochastic game case, it's not known whether it can be solved in polynomial time. Hm. So there, it is a little harder as a problem, but it's, you know, not harder, not deeply harder and not harder in a way that matters in a machine learning setting. Cool. So this is really great. So let's, let's try to take this same approach and see if we can deal with general sum games. Okay.

AI. General-Sum Games

So okay, so let's think about General sum games, so not zero sum any more. But we're not, you know, restricted, it could be any kind of relationship between the two players. And so the first thing we need to do is realize well, well we can't really

do minimax here any more. Right, because that doesn't make sense. Right. That only works with zero-sum games. Well it's only, yeah. That's, well, it sort of assumes that the other player's trying to minimize my reward and that's not the, that's not the concept of the Nash equilibrium. We'd like to do something analogous and find a Nash equilibrium in this general sum setting. So what, what operator do you think we would need in this context here? Nash equilibrium? Yeah, so that would be a very reasonable thing to do, is instead of computing mini max, we actually compute of the two matrix game, right, using Q1 and Q2, compute the Nash equilibrium of that and propagate that value back. It's a well defined notion, right, that we can summarize the value of these two pay of matrices with with a pair of numbers which are the values of the Nash equilibrium. Mm-hm. Alright, so so good. So we can do the same thing in the Q learning setting. Substitute in a Nash equilibrium. And we can call that algorithm Nash-Q, which is, appears in the literature. Nice. Oh minimax Q by the way is something that I wrote about. Nash-Q is a different algorithm. So it's not as cool, is what you're saying. Well, let's let's see how it goes. So this is now an algorithm, you can actually, well, this set of equations it's not exactly clear what it means, but we can think about turning that into value iteration, right? By turning this into an assignment statement. Mm-hm. So, what happens? Well, value iteration doesn't work. No. So, yeah, so if you, you repeat this over and over again, things, weird things can happen, it doesn't, it doesn't really converge, it doesn't really solve this system of equations necessarily. Hm. And unfortunately the, the reasoning here is even harder in the case of Nash-Q because in the case of Nash-Q, it's really trying to solve this system of equations using something like value iteration, but with extra stocasticity. And so it also suffers the same problem. It doesn't necessarily converge. There's not really a unique solution to Q star because you can have different Nash equilibria that have different values. Right. So there isn't really much hope of converting to the answer because there isn't the answer. The the policies can not be computed independently, right, so Nash equilibrium is really defined as a joint behavior, and so we can't just have two different players computing Q values. Even if we could compute the Q values. It wouldn't necessarily tell us what to do with the policies, because if you take two different policies that are both half of a Nash equilibrium, two halves of a Nash equilibrium do not necessarily make a whole Nash equilibrium. Right. because they could be incompatible. So, you know, so far so good, right? Yeah, I can't wait to see what happens next. The update is not efficient unless P equals PPAD, which is to say, computing a Nash equilibrium is not a polynomial time operation as far as we know. It is as hard as any problem in a class that's known as PPAD. And this is actually a relatively recent result, in the, in the last five, ten years. And this class is believed to be as hard as, as NP. So, possibly harder. So it doesn't really, doesn't really give us any leverage to, computational leverage to kind of break it down in this way. So that's unfortunate. And finally, the last little hope of, well, maybe we can define this kind of learning scenario using Q functions the same way we've been doing, Q functions are not sufficient to specify the policy. That is to say, even if I could do all these other things, efficiently compute a solution of, you know, build the Q values, make them so that they're compatible with each other. And now I just tell you, here's your Q function. Now decide how to behave, you can't. It's, there's not enough information. You're depressing me, Michael. Yes, so this is kind of sad. We go to the general sum case, which in some sense is the only case that matters' because zero sum never really happens. And what we discover is that we lose all, seemingly lose all of the leverage that we have in the context of Q type algorithms. Mm, mm, mm. And that's where we'll stop. Oh. So we're going to end on a high note. No, maybe we should say something before we depart. Let's do that. Come up with something positive to say. Okay.

AJ. Lots of Ideas

So even though things are kind of grim, with regard to solving the general sum games. There are lots of ideas that, that have proven themselves to be pretty useful for addressing this, this class of games. It is not the case that any one of them has, has emerged as the dominant view, but, but these are all really cool ideas. So here's one. You can think about stochastic games as themselves being repeated. So, repeated stochastic games. We're going to play a stochastic game and when it's over, we're going to play it again. And that allows us to build folk theorem-like ideas at the level of stochastic games. Oh, that's cool. And so there are some efficient algorithms for dealing with that. So that's one idea. Another one is to make use of a little bit of communication side-channel to be able to say, hey, other player. Here's this thing that I'm thinking about. And it's cheap talk in a sense that, it's nothing that's being said is binding in any way but it gives the two players the ability to co, to coordinate a little bit. And you can actually ultimately compute a correlated equilibrium, which is a, a version of a Nash equilibrium that you know, requires just a, a little bit of coordination, but can be much more efficient to compute. And you can actually get a near optimal approximations of the solution to stochastic games using that idea. Yeah, that's cool. Didn't, didn't I do some work in this space? You did. That's where I got the idea from. Oh okay. There's some, some work by Amy Greenwald looking at how correlated equilibria play into stochastic games and then your, your student Liam and you developed a, a really cool algorithm that actually probably approximates, the solutions. Nice. Another idea that I've heard a lot about lately, that I really like, is the notion of a cognitive hierarchy. The idea that what you're going to do is instead of trying to solve for an equilibrium, you think about each player as assuming that the other players have somewhat more limited computational resources than they do. And then taking a best response to what they believe the other players are going to do. This turns out to be a really good model of how people actually play when, when you ask them to do games like this in the laboratory. Huh. Yeah, the good news about this idea is that, because they're best responses, they can be more easily computed. That, that it's more like, cue learning in MDPs again because you're assuming that the other player is, is fixed. Okay. I'll buy that. And, the last idea I want to throw out is the notion of actually using side payments so that the players, as they're playing together, cannot only take joint actions, but they can say, hey, I'll give, I'm going to get a lot, but if we take this action, I'm going to get a lot of reward. I'm going to give some of that reward back to you, and that will maybe encourage you to take the action that I need you to take so that we'll both do better. And so there's this lovely theory by a father and son duo that they call coco values. Coco sounds awesome but it stands for Cooperative competitive values [CROSSTALK] and so it actually balances the zero sum aspect of games with the mutual benefit aspect of games. So it's, it's, it's a really elegant idea. So basically, the problem isn't solved but there are a lot of cool ideas that are getting us close to

solving it. That's right. Yeah. So even though the one player and the zero sum cases are pretty well understood at this point, the general sum case is not as well understood. But there's a lot of really creative ways that people are trying to address it. So, that is good news.

AK. What Have We Learned

Okay Charles, what have we learned? And I mean specifically in the context of this game theory two lesson. That's a that's a good question. We learned about Iterated Prisoners Dilemma. Which turns out to be cool, and it solves the problem, and we learned about how we can connect iterated prison's dilemma to reinforcement learning. What do you mean? Through the discount. Yeah, so I think of that as being the idea of repeated games. Right. Let's see, what else have we learned? So we learned about iterated prison's dilemma, which allowed us to get past this really scary thing with repeated games, connected it with reinforcement learning. The discounting. And then we learned other things like for example, I don't remember. What, what did we learn? Well so the, the connection between iterated prisoner's dilemma and repeated games was the idea that we can actually encourage cooperation. And in fact, there's a whole bunch of new Nash equilibria that appear when you work in repeated games. That was the concept of the Folk Theorem. Right. The Folk Theorem. So, the Folk Theorem is really cool. And this whole notion of repeated games really seems like a clever way of getting out of what appear to be limitations in game theory. Right. Yeah. And in particular by using things like threats. Right. But only plausible threats. Right, so that was the next thing we talked about. The idea that an equilibrium could be subgame perfect or not, and if it wasn't then the threats could be implausible. But in the subgame perfect setting, they're more plausible. Right let's see and then we learned about Min-max Q. Well there was one last thing we did on the repeated games, which was the Computational Folk theorem. Yes you're right. So basically what we learned is that Michael Littman does cool stuff in game theory. Or at least he does stuff that he's willing to talk about in a MOOC. Yes, so that's, that's there's actually a technical term for that right? MOOC acceptable research? Oh, I didn't know that. Mm-hm. So all these things are by virtue of the fact that they showed up in this class look acceptable. Exactly. Alright, you're right, but then we switch to stochastic games. Mm-hm. And they generalize MDP's and repeated games. Mm-hm. Anything else? Well, that particularly got us to min-max Q and then eventually to Nash Q. But despite the fact that Nash Q doesn't work, we ended up in a place of hope. [LAUGH] We end with some hopefulness. Yeah, and you know, I think that that's actually a lesson for the entire course. That at the end of the day, sometimes it doesn't always work, but there is always hope. [LAUGH] We don't give up and that's, that's, that's how research works. Even when we have impossibility results for things like clustering, or multi-agent multi-agent learning and decision making, we still keep struggling forward. And keep learning, and isn't that what's really important. I think so. Its important for us, and its important for machines. Yes, that is beautiful. I feel like we've made it to a good place, Michael. Perhaps we should stop. [LAUGH] Well it has been, it has been delightful getting to talk to everyone, and it has been very fun getting to talk with you, Charles. And thanks to everybody for making this happen. I agree. And we have one more chance to talk with one another as we wrap up the class. And I look forward to that. So, I will see you then, Michael. Awesome. Do we get to see each other in person for that? We get to see each other in person for that. That will be fun. Yay! Okay, well, bye, Michael. I'll see you next time. Bye. See yeah. Bye, bye.