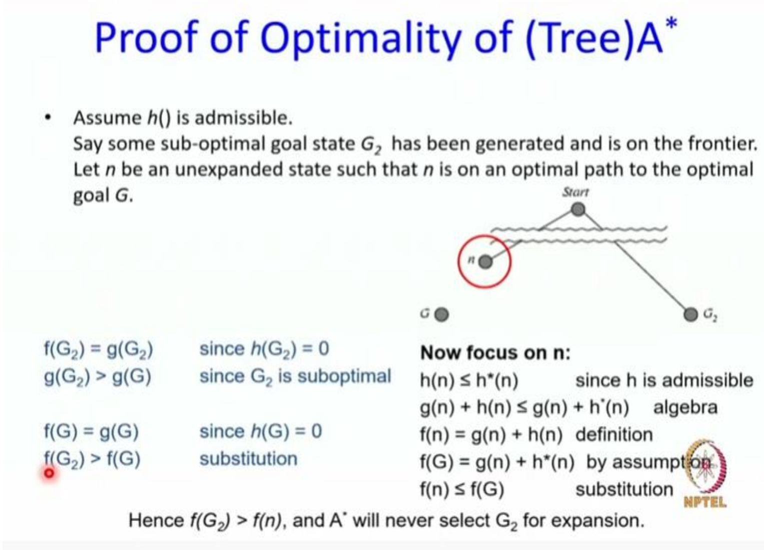
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering
Indian Institute of Technology - Delhi

Lecture – 18
Informed Search: Proof of Optimality of A* (Part- 4)

Let us do one proof. I have promised you that in this class we will not do too many proofs, but we will do some very few. So let us do one proof and the proof that I am going to do today is the proof of optimality of tree A star. So the statement is that if my h is admissible then whenever A star removes goal from the fringe, we would have found the optimal path.

(Refer Slide Time: 00:44)



So now for the next 10 minutes we are trying to just look at how to prove optimality for A star algorithm when the heuristic is admissible. So if heuristic is admissible and let us suppose that we are about to the remove G2 from the fringe, G2 is one of the goals, one of the goal nodes. So we believe that the algorithm should not finish, and we have found the optimal path. But of course let us say that it is not really the optimal path.

That means there is another path which is the optimal path. And if it is another path it would have its own goal node because its free search so let us say the optimal goal nor that we are looking for is G. And either G may be in the fringe or some other node to that path has to be in the fringe at this point in time. Okay so let us make sure that we are comfortable with this setting, right? We have a start node S we are about to remove G2 from the fringe. G2 is in the fringe we are about to remove it.

We have found we believe we have the algorithm should stop now and we are found an optimal path to the goal and let us suppose we have not. If we have not that means we are looking for a different path and if you are looking for a different path, then it has to differ at some point and whenever it differs it will create its own nodes.

Therefore, the goal node that we are interested in is a different goal node. Goal node that we are interested in is G, but we are about to remove due to no. If the goal node that were interested in G that means, there is some best path from start to G there is some optimal path, right? That is some path from start to G. let us say that path is this one on the in the diagram and let us say that we have to have some node in the fringe that leads us to this goal let us say that node is n okay. But we chose to expand G2 we chose to remove G2 and not remove n.

So if the algorithm was right it should have removed n, but the algorithm tried to remove G2. That means for whatever reason the algorithm thought that G2 is the right node to remove that means its F value was less than hence F value okay. So let us first see what we can prove about things. So the first thing G2 is a goal. That means its heuristic function is 0 that means its F value is same as its G value. That is a simple thing we can say f of G2 = g of G2.

The second thing we know is that there is a contradiction. It is not the optimal goal. If it is not the optimal goal that means g of G2 right the start state from G2 is greater than paths from start state to goal. That means that g of G2 is greater than g of G. So far so good now if you look at G if you focus on G what can we say? Well we can say that Oh its also a goal, so h is also 0. That means S its F is equal to its G okay.

And moreover because g of G2 is greater than g of G and f of G2= g of G2 and f of G = g of G. therefore we can combine these three facts to say that f of G2 is greater than f of G. they are not saying much here. Were basically saying that the f function for G2 is greater than f function for

G because we have assumed that G2 two is not the right goal and etches are 0 anyway so that is so far so good.

Okay so far okay so now we are going to focus on n, right? Because this is the node in the fringe that we chose not to expand and the first thing we know is that h is admissible means h of n is less than equal to h star of n. Okay h star of n it basically means the best path from n to goal and best path from n to goal would be n to G obviously. Now let us add the gn on both sides. So we can say that gn + hn is less than equal to gn + h star n which basically means gn + hn is what f of n right and what is f of n, f of n is gn + hn so we can replace that in the inequality.

And now let us look at gn +h star of n, what is gn + h star of n it is the cost to n and the optimal cost to g right. Now if I know that the path to G is the shortest path. Yeah this is the step only one step is interesting here. The path to g is the shortest path so what is going to happen? Fn sorry gn + h star n which is the cost to n so far and the best cost to a goal is going to be exactly this paths cost alright? This is the optimal path so therefore they can say that f of G, right?

What was f of G? The cost of this path plus 0 is going to be exactly equal to the cost of the path up to n plus the optimal path from n down the line to G. so then basically we can replace in the second equation which is gn + hn less than equal to gn + h star n we can say that left hand side is nothing but fn and the right hand side is nothing but g of G.

So basically, they can say that f of n should be less than equal to f of G? Now what are the things that we know? We know that f of n is less than equal to f of G and f of G is less than f of G2 that means f of n is less than f of g2, if f of n is less than f of G2 what should my A star algorithm have done? It should have expanded and, but it expanded G2 therefore there is a contradiction. That is my proof it is really a one slide proof.

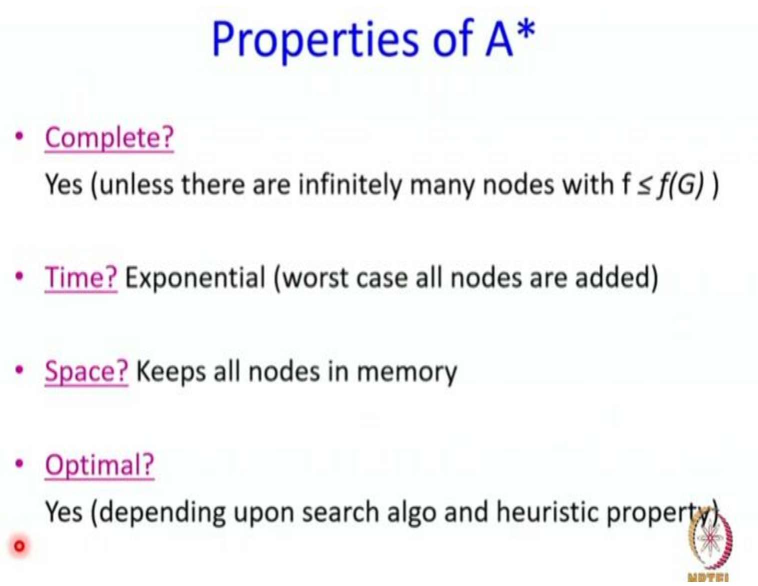
Okay so they induced any the thing is because h is admissible, I am being optimistic for the future so whatever nor looks better to me in the full path estimated length. Once I expand this I will always be staying under the optimal path and so when I finally they moved the goal. I would have found the optimal path because every other path will have a higher F value and that higher f value would be lower than the optimal path length to the goal through that particular Sub path and therefore I would have found the optimum.

This intuition of optimism is extremely important and its going to come up again and again in a variety of places not necessarily in the scores but in all kinds of search algorithms. The idea that you should be optimistic is a great idea. Later we will have uncertainty we will have probabilistic models and there is a concept called optimism in the face of uncertainty that if you can have you know range from negative to positive and you are maximizing take the positive as its value that is going to help you in getting to the optimal path.

So this intuition that admissibility optimism is awesome for search and remains with you in a variety of search algorithms. And the proofs are sort of very similar because once I find a solution the solutions that I have ignored can be proven to be worse because I was being optimistic throughout once I reach a real solution and this year solution is better than the others and others are optimistic. Their solutions are going to be even worse. Right? So then what I found is the best okay.

So I encourage you to think about this proof and also read up the proof for A star graph search for consistency. And you know you will get some intuition about how to prove this optimality for these search algorithms.

(Refer Slide Time: 10:36)



So what are the properties of A star? So we will not worry about time and space as I said because time and space is going to be exponential. You will have to if your heuristic is terrible you in the worst case, I can get really bad and I as I said if you can prove terms right like if your heuristic is so much worse than optimal then your time complexity can be better. In fact, it can even the polynomial if the divergence from the optimal heuristic is you know bounded by log log of h star or something like that and there are some proofs like that.

So you can check but we will just worry not worry about it. We will say it is exponential. Now when will A star not be complete? This is an interesting question what needs to happen in my search space such that I get into some some other territory and I never come out of it. I just never come out of it and I never find a goal.

What needs to happen I am expanding in F order. I have an optimal cost to the goal. What needs to happen no we will not worry about negative values. Negative values is a whole another ballgame, completely right? That let us just leave that for minor exams. So essentially intuitively or what is going on I am always expanding the lowest F value node and eventually when they expand goal I am done. What needs to happen that I never expand goal.

“Professor - student conversation starts” No you were saying geometric progression but what is geometric progression got to do with it? You have to say in terms of the properties algorithm yeah. So the suggestion is in my front I already always have suboptimal goals and all the costs for them are same and you should always expand the lowest F value and they will always be one of the nodes which will lead to the optimal goal and that will have lower F, so we will expand that.

We just proved this right what did we prove that we will always have another node n which will lead me to the best solution and its F value will be lower. So we will expand that so that particular case is not possible. Yes, what is your name? Rajveer yes okay very good. So the geometric progression idea comes back in, but the suggestion is that let us say my goal is at a distance 2.5 and let us say I am I have a sequence of nodes which have geometric progression of 1then half then 1/4 then 1/8 I will keep going "**Professor - student conversation ends"**.

And let us say all of them have heuristic 0 which is admissible then what is going to happen eventually I will hit the goal? No because its heuristic is higher F is higher and I will have lots of nodes whose F value will be lower than the goal, but they are never finishing. So what is the generalization of this? Geometric progression is one specific example, but I need infinite nodes of what should be the property of those infinite nodes?

See what am I using? I am using the F value. Their F value needs to be lower than which minimum value. So we have the optimal cost path that we are looking for let us say that value is f of g we are looking for a path of length f of g the goal the optimal goal. But we may not find that path if I have infinite nodes whose F value is less than that. Exactly. Okay so you will get some intuitions of this algorithm slowly right it takes time.

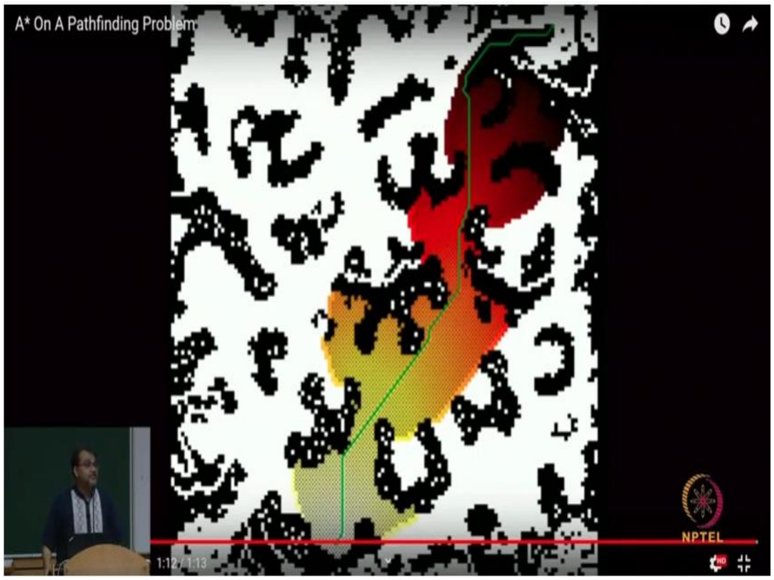
So you will have if you have infinite many nodes with f less than equal to f of G actually f less than f of G. well that depends on how you do tie breaking. So let us just do f less than equal to f of G. So if you have infinite nodes with f less than equal to f of G then I will never get to goal because I will keep adding them and I will keep expanding them and never get to the goal. Right? And that is possible with this example of geometric progression.

So if that does not happen then at some point all the nodes which have value less than equal to f of G would have been expanded and I will get to the goal node and then it will be complete. Right? Again we are not talking about negative costs or 0 costs. Okay and would it be optimal? We have already done it will be optimal based on the heuristic property and search property. So tree search admissibility or graph search consistence.

So these are the properties of the algorithm. This is considered to be one of the most important algorithms. Traditionally one of the most important search algorithms traditionally. The A star algorithm was developed as path of the shaky project in Stanford and it had far reaching consequences. And in fact it is said that Bings driving directions in Bing travel like Google maps we have Bing maps at least that I read a report about 10 years ago was using some notion of hierarchical version of A star. And A star has been used extensively in a viral idea of algorithms and continues to be used.

So it is well the algorithms in the 60s which is still as important as anything else okay in here. And now let us see how the run is when we tried to go from the start state to the goal state.

(Refer Slide Time: 17:27)



So by the way I should say that these beautiful videos are courtesy university of New Hampshire and my good friend and colleague or at least research colleague Wheeler Ruml who is a student realize that he is a well-known expert in search algorithms. And notice let us stop for a second.

Notice how fast it is moving towards the goal. If you remember from last class, you know depth first search will do completely random things breadth for search will just go all around and you know look everywhere in the world basically in all directions he had noticed that I am giving it a heuristic function.

The heuristic function is a straight line distance, right? So notice how quickly it starts to go towards the goal. Now you will be dissatisfied in the middle and we will talk about why. So let us see how this algorithm works. Now are you a little dissatisfied at this point? What is the algorithm trying to do? Why is it searching for other paths on the right and so on and so forth when it feels as if it is so close to the goal. It knows its very close to the goal.

These points must have relatively less distance to the goal. Why is it not moving ahead? because it also needs to prove. See the G here is fairly high. The cost from this node to this nodes very close to the green dot.is actually fairly high. So it now needs to prove that there is no alternative path that takes me to the goal faster. So it will keep moving those node suboptimal and only when all the nodes other paths are you know looking worse only then it will expand the right path.

So it will take some time and notice now its sort of almost found the path and it has now found the path, right? So let us look at the final paths that it found. Oh its I have to stop it at the right place in order to do that. Right? So now it found the path it found the optimal path. Now the beauty is that is look at the beauty first. The beauty is imagine depth for search. It had expanded so much, and it had not expanded. Everything of course gone down that hole and finally came out from the wrong side and finally reach the goal. It had resulted in terrible path.

So we do not want that kind of thing as our intelligence. We do not want the algorithm to just keep going and not the algorithm the robot or any agent to just seemingly wander aimlessly. And then finally randomly we hit the goal that is not very good. Breadth for search would have found the same path but breadth fort search would have expanded almost every node in the graph. And that was terrible because it was looking in all directions. It had no idea where the goal is.

There was no notion of you know look ahead. There was no notion of estimates of the future. There was no understanding of the future. It was just blind, so it was looking in all directions equally. Here the A star algorithm expanded a far less search space than breadth for search and still found the optimal solution. Now of course if you did greedy breadth for search and if it is not stuck in loops and so on and so forth it would be even more goal directed.

It will say you are closer to goal I will go to you. You are closer to goal I will go to you with complete disregard on um what has happened so far. So it would expand even fewer nodes if it hits the goal. But it will not be guaranteed to be optimal. It will not be guaranteed get into to actually reach the goal. Okay so A star is a great balance between optimality and trying to use the heuristic function effectively. Here is one important thing that I missed talking about.

There is a heuristic function and there is a heuristic and those two things are different things. When we use the word heuristic function it is a very defined vertical concept. We are talking about the estimate of the node to a goal and so on. When I say it is a heuristic strategy when I say it is a heuristic algorithm, when I say it is a heuristic, I am basically saying I cannot prove much but it is a nice trick to add in my algorithm. That is the notion of a heuristic there is an intuition as to why I am adding it but even if I do not have it the algorithm it may might work. If I add it much better but I cannot do much here. That is the indication of a heuristic.

And do you know about HAL like from the space odyssey do you guys watch science fiction? Do we know what HAL stands for? It stands for heuristic algorithmic. Some people thought that it stands for IBM because H is one letter less than A is one letter less than B. And L is one letter less than M but that was just a fluke. Hal in space Odyssey had nothing to do with IBM. It stood for basically heuristic algorithms.

So in AI as we like our terms, we also like our theorems we also like our heuristics. So in the field of theory heuristic maybe a bad word you will say this fellow was not smart enough to prove anything. This fellow came over heuristics to make it work. But in AI we are happy about our existence we feel like if we added an insightfully the stick which made the algorithm go much faster. In fact, this even if we cannot prove about it great awesome.

Okay. But when they use the word heuristic function or heuristic in the context of A star that is a different thing. There is a theory behind it. It has a particular meaning it is an evaluation function of a sort. If it has certain property, we can prove things about it etc. So that is a different, so people confuse heuristic and heuristic function. So I thought I will clear it. We can stop here. Thank you.