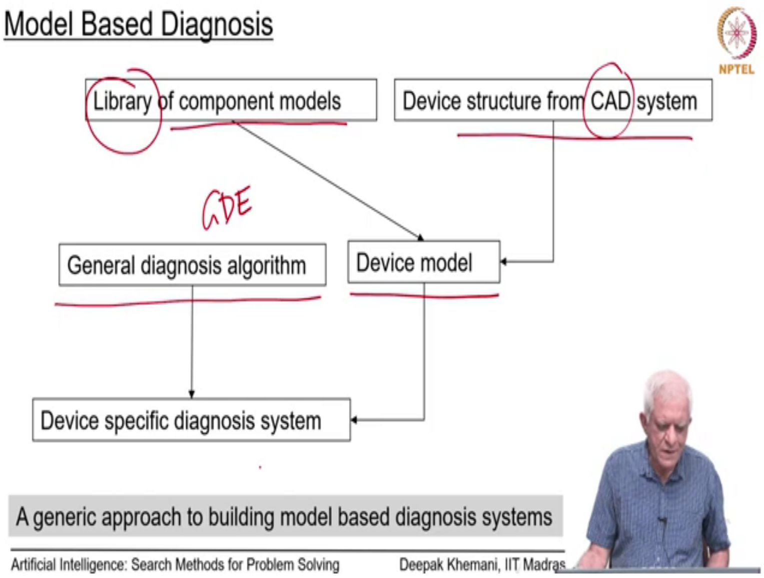
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 09
A First Course in Artificial Intelligence
Lecture – 91
Constraint Processing
Model Based Diagnosis

(Refer Slide Time: 00:15)



Let us look at one more example before we focus on the algorithms. And this example as I mentioned in closing in the last video is a very interesting application of using constraints. And this application is that of diagnosis.

Now, diagnosis is a big problem and we had mentioned earlier, when we talked about rule based expert systems that you can you know extract rules from people and experts. And then, use those rules to do diagnosis and in fact, one of the programs that was done for diagnosis called Mycin was doing something like that.

But that was a knowledge-based approach, that was in some sense, you might say superficial knowledge. In the sense that the program did not have a model of the object, it was trying to diagnose or the entity that it was trying to diagnose.

But worked with heuristic knowledge that if you for example, if you have these symptoms, then you have this disease likely and so on and so forth. Model-based diagnosis takes a more first principles approach.

So, it is in that sense, more consistent with the what we are doing in this entire course, which is like search is like a first principles approach. You give us a new problem and we will use search to try and solve it essentially. In a knowledge-based approach, if you give similar problems, then we will be able to solve those problems. Or if you have somehow managed to acquire knowledge about a particular class of problems, we will be able to apply that knowledge.

So, it is not that one is better than the other. Eventually of course, you will have to do a combination of things; because using knowledge-based systems is going to be faster than search because we have been talking about all this while about this combinatorial explosion.

So, it is a interesting landscape to navigate, if you are trying solving hard problems. So, model-based diagnosis takes a first principle approach and it works as follows that you first describe component models. So, if you have a device which is made up of components, you first model the component independently of the role, it is going to play in the larger device.

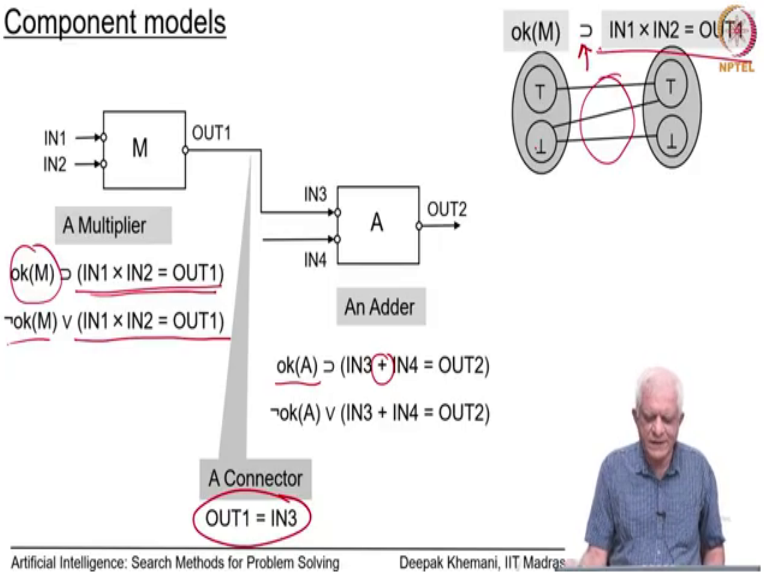
So, the community, the model-based diagnosis community calls it no function in structure essentially. That the structure that we get about how components are put together, do not change the function of the component essentially. This is the assumption that they work with essentially.

That the component behavior can be described independently of where they fit in into the larger structure and the idea behind model-based diagnosis or model-based reasoning is to work with a library of component models, take a drawing like from a computer aided grafting system and what you get is a device model essentially.

So, the device model gives you components which these are components that are being used and this is the structure of how they are connected together. Once you have a model, you use a general diagnosis algorithm. So, some people call it a general diagnosis engine GDE. So, you can just look up for GDE here and you will.

So, it is in the same spirit as our entire course that our problem-solving algorithms are general; the problems have to be plugged in essentially and what you get if you do all this is a device specific diagnosis system. It is a generic approach to model-based diagnosis essentially. We will just try and get a flavor very quickly about this. We do not have time to go into the details and there are there is a considerable amount of detail here.

(Refer Slide Time: 04:26)



So, let us talk about component models first ok. So, and we are talking about constraints. So, we should try to see everything from the perspective of constraints. If I have a model of a component and remember that, I am talking about we are talking about diagnosis. So, we will use a label like ‘ok or not ok’.

So, we will use two-two reverse labels; one says M; M is a component and M says that its ok, it is not broken, it is not faulty and if you in this case, I am talking about a multiplier here. So, this is a component, we are talking about; its a multiplier and it can be described by expression that the input is related to the output by a product relation essentially. That if the device is ok, then this is a true statement.

This is expressed in logic and equivalently, you could have said not M or this statement. As you know a little bit of logic we have studied and these two statements are equivalent states.

What I want to start off by emphasizing the fact that this logic relation, remember logic is based on relations and constraints are also based on relation.

So, it is not should not be a surprise that you know there is connection between these two ways of looking at things. So, in particular, we are looking at the implication relation here. There are two statements the antecedent says M, the multiplier is ok; the consequent says that the product of the two input is the output essentially.

So, if you focus on the implication relation, you can see that it is a its a constraint when treating M as one variable and this other sentence the whole thing has another variable and this is basically the truth table of implication. Remember that these are the three rows in the truth table; true, true, false, true, false, false for which implication is true.

So, implication is expressed as a constraint here and this is how we will see models in this very simple example that we are seeing essentially. So, what are we saying here? We are saying that a multiplier is or if the multiplier is ok. This is what we are saying here, then the output is related to the input as a product of the input essentially. Then, in a similar fashion, we can describe an adder, that an adder is ok, if the two inputs add up to the output.

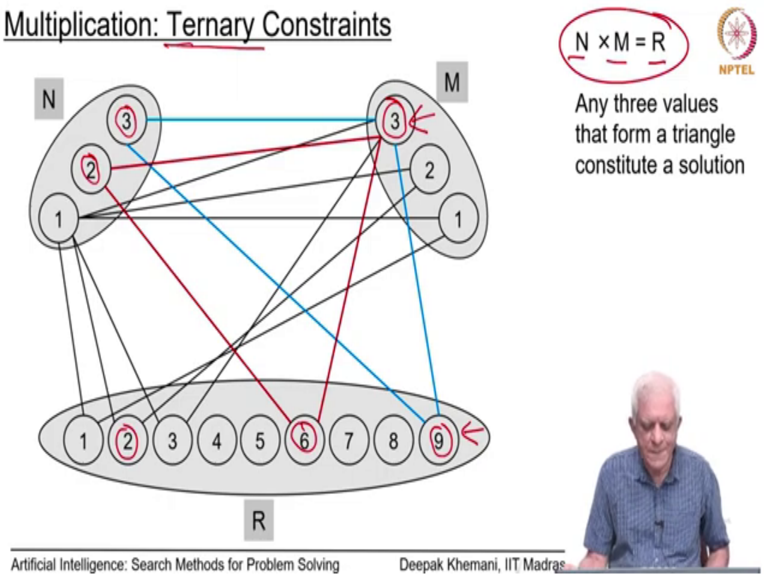
So, the relation is add here. It is contingent on the fact that the device the component is ok. Remember, we are looking at the problem of diagnosis and the way that we are posing this problem now is saying that diagnosis is the task of first identifying that there is something wrong with the device and secondly, identifying which component has broken.

So, it is a very simplistic approach to diagnosis. It says that only components can break and nothing else can go wrong with the device, what else? You can have connectors essentially. So, you can say that the output of this multiplier goes as an input to the adder, one of the two inputs to the adder and this is expressed as another constraint which says out one equal to in three in this diagram.

So, using components and connectors, so we have component models. We have a model for a multiplier, we have a model for an adder and then, we can put together adders and multipliers to create some interesting devices and those will be defined by the structure. The structure is given by connectors essentially and our simplistic assumption is that only components can go wrong. So, we cannot say that this wire is broken or something like that, you know that is not part of our definition of the problem ok.

So, we have this the key thing that I said here was that we can describe model behavior, component behavior using constraints, that is the key thing. So, the statements themselves which are logical statements because we are expressing in one sense in logic M or IN 1 into IN 2 is equal to OUT 1 are either true or false. So, they are expressing let us say first order logic or something like that; but they themselves can be expressed as constraints.

(Refer Slide Time: 09:44)



So, let us look at multiplication as an illustration of the fact, that we can talk about that as constraint. It is not a surprise because you know logic mathematics constraints, they are all different sides of the same coin, if a coin can have more than one side. So, here is here is an example of how you would represent multiplication as constraints.

So, again, I must emphasize that we do not do that you know because it is a very painstaking way of doing that; but from the first principles perspective we can think of it as a constraint essentially. We may not actually implement it like that, but if you wanted to really be a purest constrained satisfaction problem researcher, then you might want to do this.

So, multiplication is a ternary constraint. Why because it is a constraint? It is a relation between three elements; the two inputs N and M and one output R and this is the matching diagram that we have seen. We have seen a taken a very small domain, where the domain of N is just three numbers; 1, 2, 3. The domain of M is also the three numbers; 1, 2, 3.

And the domain of R is numbers 1 to 9 because that is the maximum we can get with those 1, 2, 3. I have not drawn all the complete matching diagram, I have drawn only some of them. I simply want to highlight the fact that the solution to this constraint satisfaction problem is going to be three edges in this matching diagram. I have given two examples here; one is in red which says that 2 into 3 is equal to 6.

So, there is an edge; there are three edges between these two things. You can say that if you can find the triangle in the matching diagram, then that is a solution. Another example, I have taken here is that if you have 3 into 3, then it is equal to 9 essentially.

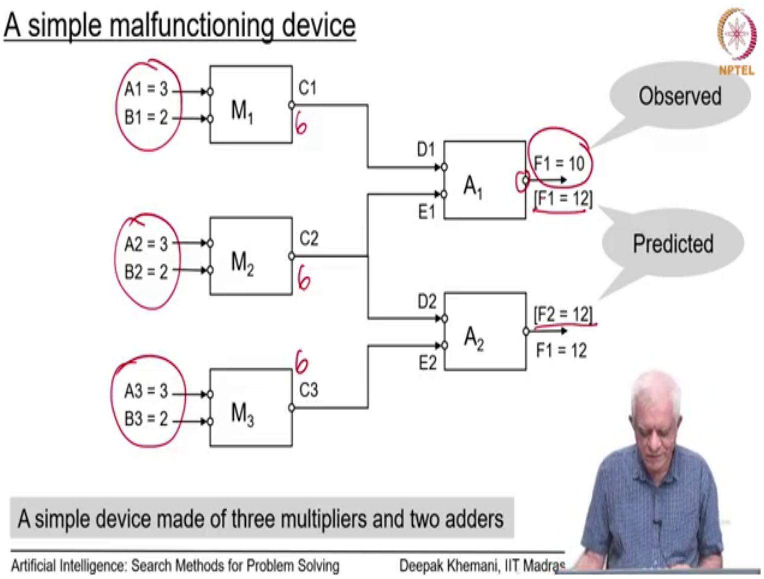
So, remember that when we said, we can solve constrained satisfaction problems. We can say give us a solution. So, this particular formulation would give us some random two numbers and their product; the nice thing about constraints and it is also true of logic programming as we had said earlier is that you can specify two of these variables.

So, you can say as I was doing a short while ago. What is the product of 3 and 3? And then, the system would say 9 is the answer essentially or I can say what is the product of 1 and 2 and the system would say the answer is 2 essentially. But I could have also done the reverse, I could have specified what number when you multiplied by 3 gives you the answer 9 and the system would say N equal to 3 essentially.

So, the solver the constraint solver can be used to do multiplication and addition, you can come up with a similar table. This is just to highlight the fact that everything can be represented in terms of constraints. In practice, we are not going to take equations like multiplication and express them and solve them using constraints.

Just as we never represent multiplication as a logical relation which can be done and solve it using theorem proving, we are not going to do that here. But in principle, everything, the whole device can be expressed as a set of constraints that is a take away ok.

(Refer Slide Time: 13:21)



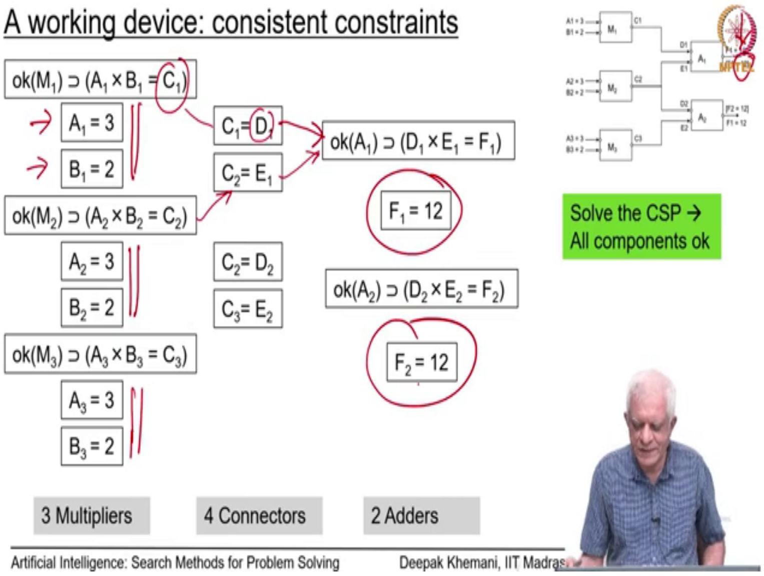
So, what device are we talking about? Here is a standard example, you can find it in almost every textbook on model-based diagnosis and this is a simple malfunctioning device. As you can see there are three multipliers; M 1, M 2 and M 3 on the left. Each of them has got two inputs and they happen to be same in this case 3 and 2, 3 and 2, 3 and 2.

The 3 multipliers feed into 2 adders as you can see M 1 and M 2 feed it into feed into A 1 and M 2 and M 3 feed into A 2. And the interesting part is you have supplied the input, which are these values and you expect a certain output which is the predicted values, which is that you expect to see 12 and 12 at the two output terminals.

These things are also called terminals; but when you switch on the device or whatever, you see something else. Observed value is 10 essentially. So, clearly the device is not working because what you should really get is the output of the multiplier should be 6 in the three places.

And then, 6 plus 6 should give you 12 and that is what the two adders should be doing; but one of the adders A 1 is saying that output is 10. So, the question is what is happening ok.

(Refer Slide Time: 14:51)



So, let us first express the fact that these are constraints and we have described these constraints, let us just put them on one page together. So, how is the whole device represented? The whole device is represented as a set of constraints. So, we have these three multipliers; M 1, M 2 and M 3 and each of them is described by the same constraint that we had seen. The inputs also expressed as constraints as you can see here A 1 is equal to 3, B 1 equal to 2 and so on and so forth.

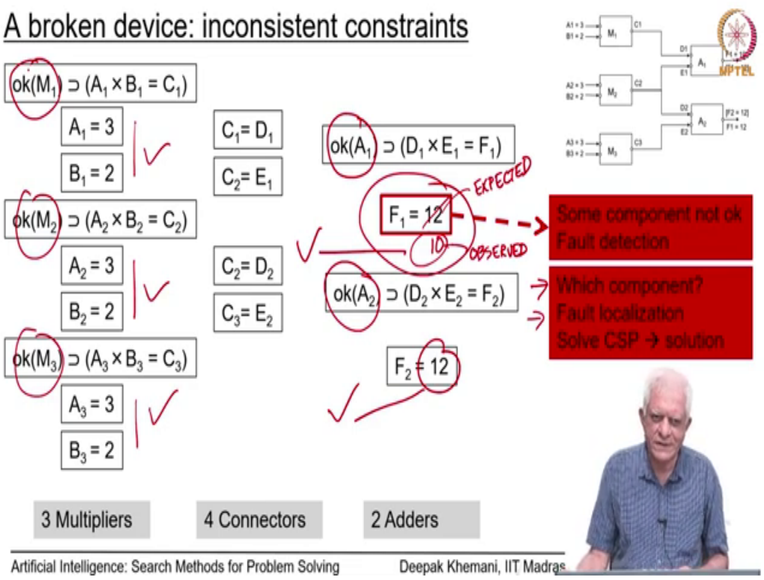
Then, we have the two adders which are described in a similar way by the two constraints for the two adders. And then, we have the connectors which says that the output of C 1, output of this multiplier goes as input to this adder essentially and the output of this multiplier goes as input to this adder essentially, that is what we are seeing.

And then, this is a the constraint system, if you solve it. So, remember that, apart from the fact that we have described the device model, we have also described the input to the model here.

And if you solve it, what you will get is this two values for the output is that F 1 should be 12 and F 2 should be 12. Now, the problem that we were looking at was that it should have been 12, but it is actually 10. Then, what happens essentially?

So, we go back to our constraint solving, this time the output that we are talking about is the observed output. See in this slide, we are looking at the predicted output F 1 should be 12 and F 2 should be 12, the observed output is F 1 equal to 10 and F 2 is equal to 12. So, what is happening there?

(Refer Slide Time: 16:55)



So, again, it is the same model. As you can see this will go through the same process of arriving at component descriptions, the input to the three multipliers, the adder descriptions, the two adders which are described; the connectors which are again described here and the two outputs, this should have been 10; sorry, that the observed output is 10 essentially ok.

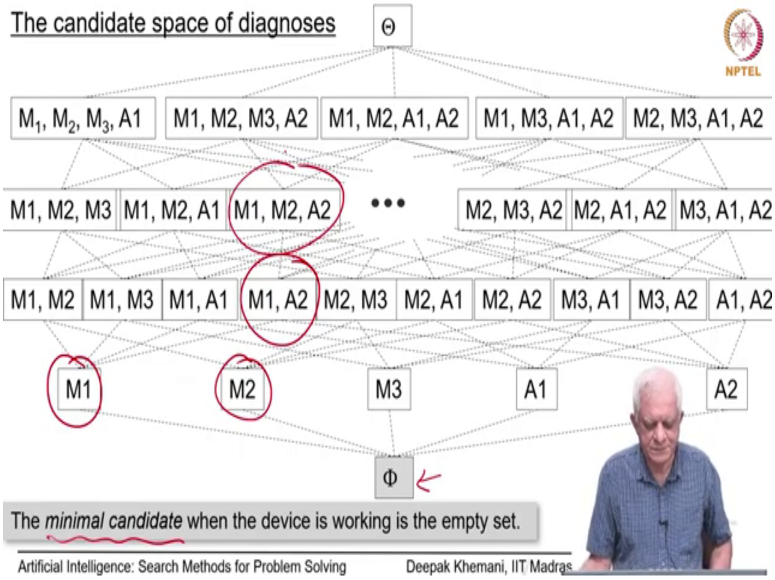
So, what happens here? That once you see that the output that there is a discrepancy between the x, the predicted value and the observed value; then, this is a discrepancy is here that the expected was 12 and the observed is 10. This 12 was expected. The perils of copy paste.

So, if the expected value does not match the observed value, then you have to do some figuring out. The first thing you must realize that some component is not ok. So, remember that we had said that our assumption is that it is the components only which can fail, the connectors are flawless.

So, we have identified that there is a fault. The next question is which component, which is often termed as fault localization and the way, we do it is that we will solve this new constraints. What is this new constraint? Where everything is given to us essentially. The two inputs are the six inputs are given to us; the observed values are given to us. So, this is also observed and now, we want to say that give us a solution for this CSP.

What can change? The only thing that can change is those statements; you said A 1 is ok, A 2 is ok, M 3 is ok, M 2 is and M 1 is ok. For this whole network to be consistent, one of those will have to change and that will be given to us in the solution and this approach to diagnosis is also called as consistency based diagnosis.

(Refer Slide Time: 19:39)



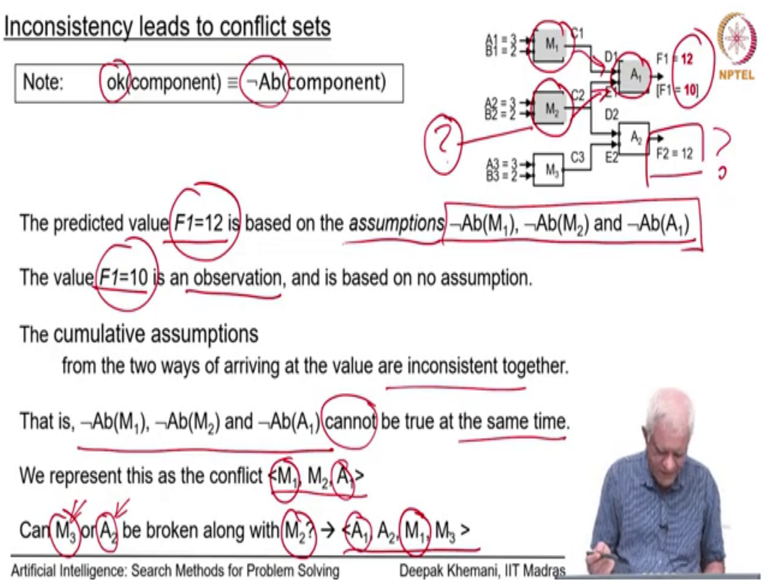
So, let us very quickly see how this is done. We start off with a candidate space of all possible diagnosis ok. So, this is just a partial order of this is just the power set of the five components. So, either nothing is faulty that is the starting point for us the and we will be interested in minimal candidate.

We can say that everything is wrong, but that does not help us, we want to localize the fault. So, whenever we have a set of candidates, we will be only interested in minimal candidates; otherwise you can just go off and say one of those five devices is gone, one of those five components is gone; but that does not help us.

We want to localize which component is failed essentially when we start the process. We assume that no component is failed till we discover an discrepancy essentially. So, eventually, we want to identify in this which component for example, is it M 2 or is it M 1 or is it a combination of two things or a combination of three things which has gone wrong?

If you can identify that, then that will be given to us by the solution of the CSV that we mentioned. Remember there are five statements; multiplier 1 is ok, multiplier 2 is ok, multiplier 3 is and adder 1 is ok, adder 2 is ok, some of these will become false and that will tell us which is the component which has failed.

(Refer Slide Time: 21:10)



Now, let us see how the community goes about doing this, very quickly. We use a different notation here abnormal. So, when we say it is ok, we mean it is not abnormal; they are the same thing. We could have used not or in one place or not abnormal in the other place; but it does not matter. Different people tend to use different notation.

So, there is a discrepancy; one is expected value was 12 and the observed value is 10; so, the predicted value was 12. And it is based on three assumptions that these three devices because the input is coming from M 1 into M from M 1 and M 2 into A 1 and going to A 1. And then, output is coming at A 1. So, if all these three devices are working perfectly, then there would be no discrepancy.

So, the assumptions are that all these devices are working perfectly which means that none of them are not, all of them are not abnormal that is the assumption. Obviously, the assumption is wrong or likely to be wrong because one of them is likely to have failed essentially. So, that is one observation that we make and this is the analysis which the community does.

The value F 1 equal to 10 is an observation, it is based on no assumption. So, the only assumptions we are making so far is that M 1, M 2 and this is done by kind of propagating these values 12 and 10 into the network and saying which components are they encountering essentially and they are encountering M 1 and M 2 and A 1. So, we say that these are the assumptions that they must be working correctly for output to have been 12 essentially.

So, the cumulative set of assumptions between both the things; the predicted value 12 and the observed value 10. So, the first predicted value analyzing the predicted value and analyzing the observed value, together they are inconsistent. So, we are saying that all of these three cannot be or cannot not be abnormal.

So, cannot be true at the same time and we express this in the terminology of model-based diagnosis as a conflict that one of these three devices at least is failed. But we can ask another question is it possible that M 3 or A 2 is wrong? Now of course, that does not sound very intuitive because along with M 2.

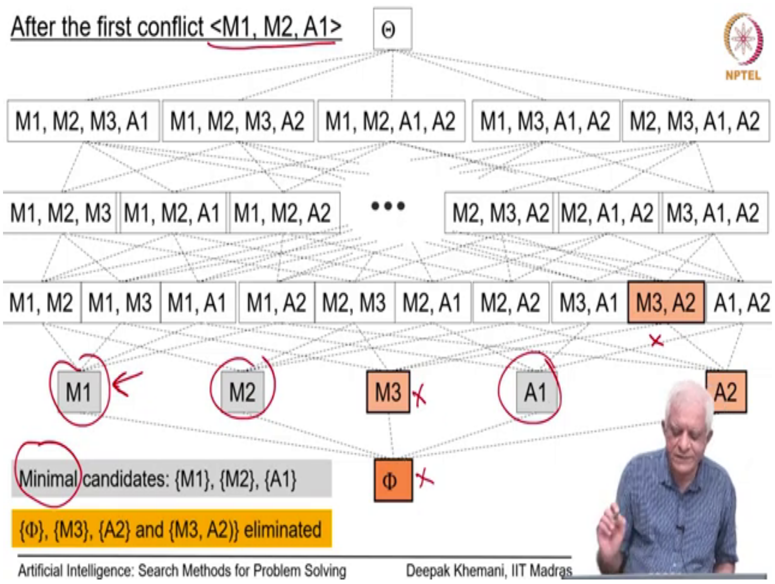
So, let us say M 2 is faulty; but if M 2 is faulty, how come we are seeing a correct output at A 2? The answer is that either M 1 either M 3 or A 3, A 2 is faulty. So, that is what we have written here. That is it possible that M 2, M 3 and A 2 are possible along with M 2.

So, because if M 2 is going to be faulty, then how do we see this value? So, there is a flow of information between the two outputs and this gives us a new conflict which is says that all these cannot be that A 1, A 2, M 1, M 3 cannot be ok. So, we have two conflicts; first one said A 1, M 2; A 1, M 1, M 2; the second one says A 1, A 2, M 1, M 2.

A diagnosis is what is called as a heating set. So, we have these two sets; is there an element which is common to both the sets? So, obviously, we can say A 1 is common to both the sets. So, it is possible that A 1 is faulty and that would explain everything that we are seeing essentially.

Alternatively, we could have said that M 1 is faulty; M 1 is there in both the conflicts and that would explain the fact that what we are seeing is explained in the sense that we can understand why the discrepancy is there. But it could also be the case that we pick M 2 from this one and either A 2 or M 3 from the second conflict.

(Refer Slide Time: 25:58)



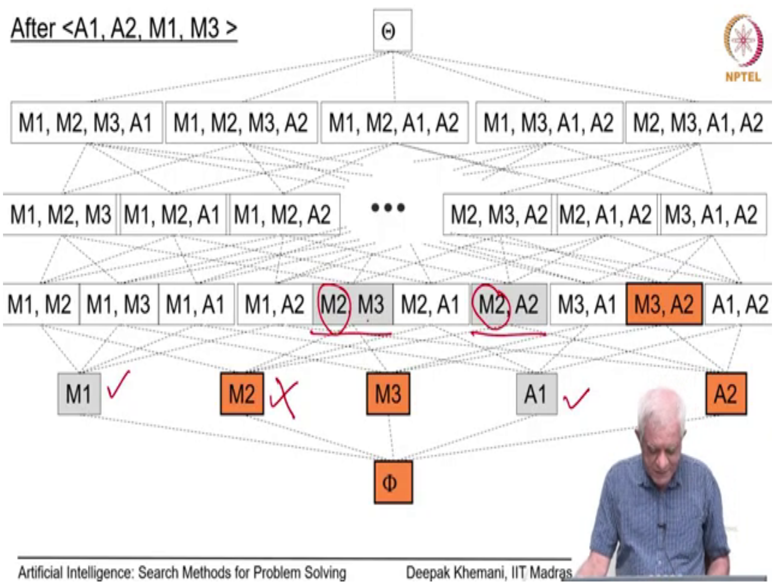
So, let us see how that happens. When we look at the first conflict, then after all this reasoning that or solving that, I have been talking about. We find that this is exonerated that definitely, there is a fault.

You cannot say that no component is gone. We can also say that this single hypothesis that only M 3 has gone is also exonerated; likewise only A 2 going bad is does not explain why A 1 was showing 10 and in a similar reason, you cannot say that both A 2 and M 3 are gone because that still does not explain why you were seeing 10 at A 1.

So, these things in this pink like color are exonerated, the other three remain as candidate diagnosis which is A 1, M 2 and M 1. Remember that this is a power set which means if A 1 is faulty, then M 1 is faulty. For example, then any superset of M 1 can be said that one of them is faulty essentially, but we are trying to do the minimal candidates.

So, we have only listed the minimal sets that are indicated by our conflict set. So, what is the conflicts that say? That either M 1 or M 2 or A 1 is faulty, which means that some things get rise above the level of suspicion.

(Refer Slide Time: 27:31)



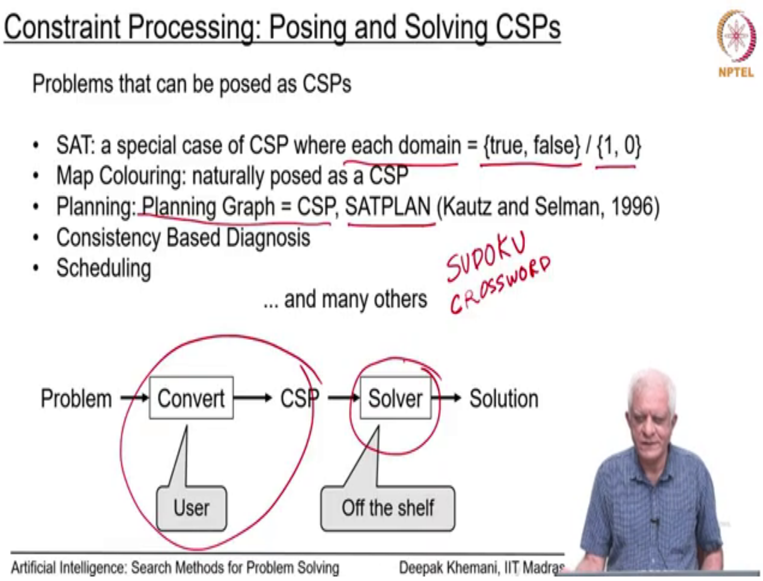
Then, we look at the second conflict which is A 1, A 2, M 1, M 3 because that is how the information flowed when we were looking at that and at this point, we say that M 2 by itself also cannot be faulty. So, that is also exonerated. So, either So, M 2 is ok, it cannot be a candidate.

So, when I put a cross, it means it is not a candidate which means its ok. M 1 can be faulty that is a single diagnosis; A 1 can be faulty that is another single diagnosis; but also its possible that M 2 and M 3 together have all gone wrong or M 2 and A 2 have together gone wrong essentially.

So, basically when M 2 is going wrong, one of the other devices that it is feeding into is in some sense correcting whatever it is undoing whatever this was doing, which means both are faulty really.

So, that is the idea behind model-based diagnosis and you can see that it is done using expressing the whole thing as a set of constraints and doing this algorithm called consistency-based diagnosis which operates on top of constraints. Unfortunately, we will have to leave it there. We should soon move to algorithms.

(Refer Slide Time: 28:41)



And this is what we have done so far. We have talked about posing problems as constraint satisfaction problems and we have seen some examples of what kind of problems can we pose as constraint satisfaction problems. So, let us just do a recap of this kind of a thing.

We have seen that SAT, we have not seen it here; but if you think about SAT, you can see that SAT is a very special case of a constraint satisfaction problems in which every domain has a value of true or false or 1 or a 0 and the relations are the logical connectives. And we have seen a glimpse of how implication can be expressed as a constraint.

So, SAT is a special case of CSP. Map coloring we have already seen, it is posed naturally as a CSP. Planning, we have not seen here; but if you remember the planning graph, then you can actually think of that as a CSP is you grow the planning graph till the condition of is met, where all the goals occur mutex free in some layer. And then, you treat that whole graph up to those that level as a constraint satisfaction problem and see if there is a solution to that.

There was another well-known algorithm called SATPLAN which used planning as satisfiability by Kautz and Selman in 1996 essentially. Or there are other things like Consistency-based diagnosis that we have just seen or scheduling which you have not seen; but you can take my word for it that. Many problems can be posed as CSPs.

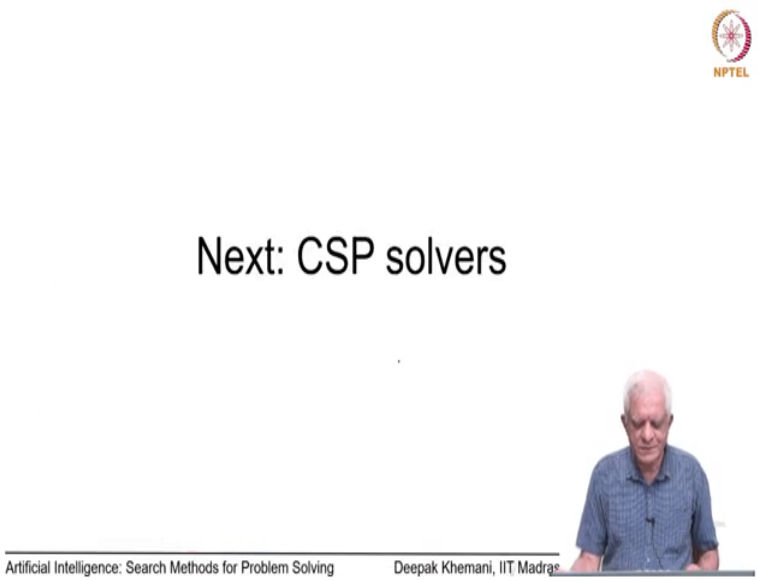
So, as an exercise, I would ask you to look at SUDOKU and see how can you express SUDOKU as a CSP essentially? Because if you think of how you solve CSPs, then there is something about the SUDOKU.

SUDOKU is almost like a map coloring problem. The two the two squares in the same column cannot have the same number for example, instead of color or you can think of a crossword puzzle, not solving a crossword puzzle; but making a crossword puzzle.

If you are an editor of a magazine or something and you want to take all the words from that particular issue of the magazine and some words from that issue of the magazine and use them to construct a crossword. Then you can think of it as a constraint satisfaction problem; give some thought to that.

So, this is a summary of what we have been talking about, that if you have a problem to solve, you pose it as a CSP and this is done by the user and we have just described how to pose it as a CSP. The next stage that we want to look at is solving them, how do you what are the algorithms, which will give you the solutions and we will do that in the next session onwards in the remainder of this week ok.

(Refer Slide Time: 32:11)



So, next we will be looking at CSP solvers.