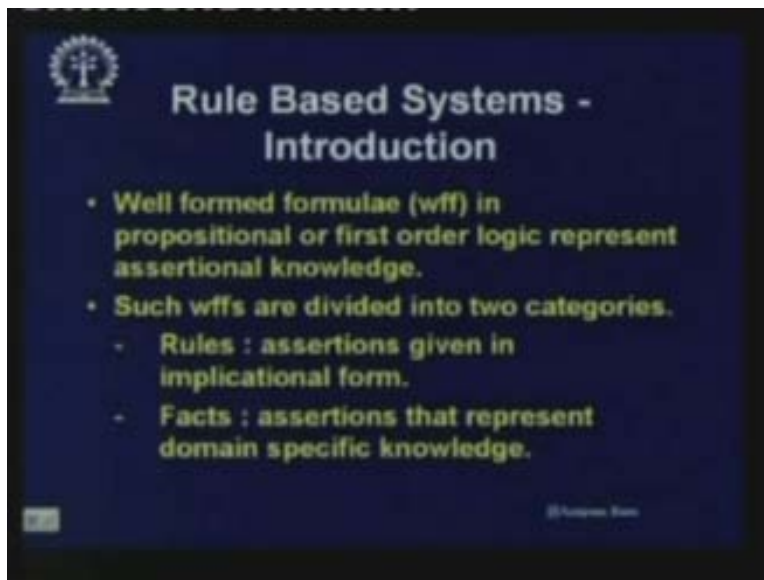


**Artificial Intelligence**  
**Prof. Anupam Basu**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**  
**Lecture - 16**  
**Rule Based Systems**

Today we will be discussing about specific practical approach to implementation of systems which behave intelligently. Earlier we discussed about one form of knowledge representation and that is logic. We have also seen how inferencing can be carried out in logic. Today our discussion will focus on rule based systems which is a very practical implementation of intelligent behavior. Of course it derives from logic. We will see how it is depended on logic and it is also some sort of simplification to make the systems implement under engineering terms. We have seen well formed formulae in the case of propositional logic and first order logic.

(Refer Slide Time 02:01)



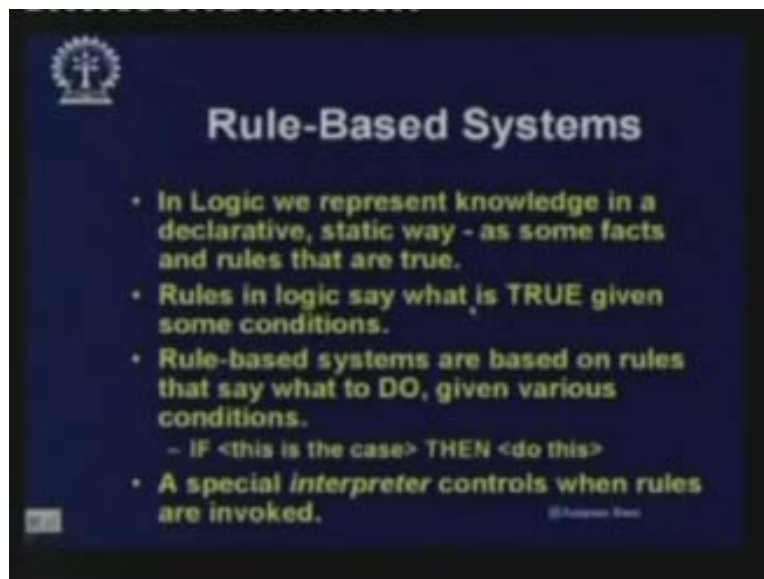
Now these essentially represent assertional knowledge or they represent assertions. What are assertions? Whenever I make a statement; today is Sunday, yesterday was very hot, these are some assertions I am making and these statements as we know from our knowledge of logic can either be true or can be false. We can say that these can be evaluated to true or can be evaluated to false these are assertions. And we have seen that propositional well formed formulae or predicate well formed formulae all evaluate to either true or false.

Now, if we look at these assertions in little more detail we can see that this can be decomposed or this can be categorized into two different forms. One is rules another is facts. Rules are assertions given in implicational form. An implication for example is,  $p$  implies  $q$  is an implication and rules are usually given in that form. There is another set of

assertions which we call facts which represent domain specific knowledge. For example, the boy is intelligent just we say the boy is intelligent that is an assertion, it may be a true statement or it might be a false statement. But when we assume that it is a fact then we usually assume that is true the boy is intelligent. But if you say if the boy is intelligent then the boy will score good marks then that is a rule we are saying and this implication is being noted by this construct if then. So we can have a complete knowledge system which consists of two distinct components. One is rules another is facts. We can have a set of facts which are presented from the domain of this course.

For example, let us consider the domain of school geometry. We can say ABC is a triangle. We can also say ABC is an isosceles triangle. AB and AC are two sides of the triangle. Now all these statements I have made are nothing but assertions but there is no implication in that. I have just said AB and AC are two sides of the triangle. I have said ABC is a triangle or ABC is an isosceles triangle. All these are facts. Now if I say, if ABC is an isosceles triangle then the sides two sides are equal now this is a rule which I am stating. And based on this rule using this implication that is inherent in this rule I can infer new facts. In logic we represent knowledge in declarative and static way. We just say we make statements which are declarative. And rules in logic say what is true given some conditions. For example, when I say if ABC is an isosceles triangle then sides AB and AC are equal. Therefore using this rule if I know that ABC is indeed an isosceles triangle then I can infer the new fact that AB and AC are equal. So rules in logic are stating or saying what is true given some conditions.

(Refer Slide Time 06:41)



What is the condition?

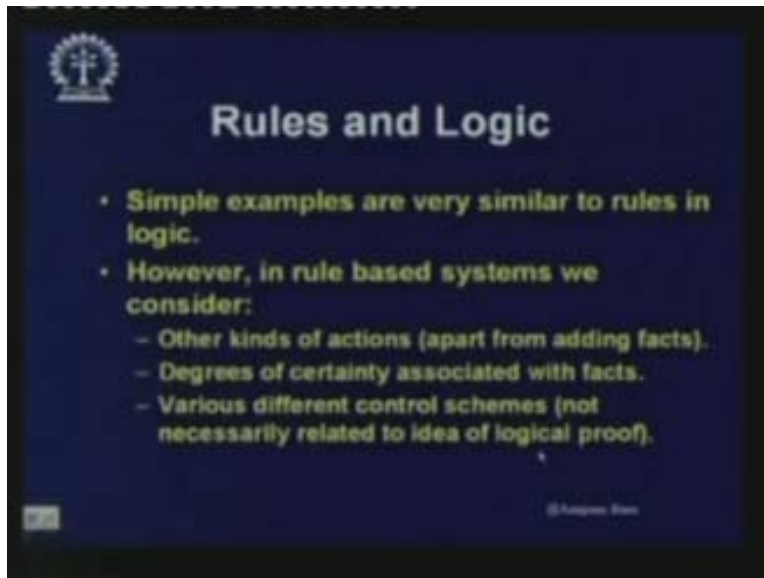
The condition is, ABC is an isosceles triangle and what am I saying using this implication? The rule is implicitly saying therefore AB and AC are equal. Rule based systems (7:03) to (7:17) are based on rules these are just implicational statement in logic. Now in case of rule based systems we go a little farther. Our implication that means the

right hand side of the implication, usually in logic makes another assertion. If the boy studies hard then he will score good marks is an assertion. But I am not asking anything to be done.

On the other hand, look at the statement: if the temperature of a furnace is very high then shut the heater off. here in the implication I am not just making an assertion I am also asking for some actions shut the heater off so that is also possible to be stated in a rule. So rule based systems are based on rules that say what to do. Here I would like to make a distinction between two types of knowledge representation. One type is declarative knowledge representation. That means as we just said what is to be done but we do not say how the thing has to be done.

Whenever we write a program for example sorting a number or write a program to draw a diagram in that program itself we specifically mention all the steps we need to carry out in order to perform a particular task. And it is only there we specify how exactly the thing has to be done. But in a declarative system as in logic we make some assertions, we make some statements. Now, which statement has to be executed at which point of time is not the headache of these rules. That headache will be transferred to another system called the control mechanism or the inference mechanism. That is why often this sort of knowledge generation is also called declarative knowledge representation whereas the type of knowledge that we are embedding in the case of a program is often called procedural knowledge representation. However in this case you can see that rule based systems can be of this form if this is the case that means the condition then do this. This is one way of statement. It is also possible to state in this way, do this.

(Refer Slide Time 10:27)



But we can also make some assertions over here and these rules are just some declarations which do not say how the thing has to be done. We require a special interpreter as a control mechanism **to know the procedure**. There are different names such

as control mechanism, inference engine, inference machine etc but the purpose is the same. This is the procedural block of knowledge that interprets and applies these rules as the case may be. Now we can have very simple rules which are very similar to the rules in logic.

What can a rule do?

A particular rule has got if you recall has got the left hand side part which we call the antecedent of the conditional part if this is the case. If it is really true that this is the case then on the other side of the implication some actions may be stated like shut off the heater, open the window, close the tap or we can make some assertions such as the boy is intelligent, the season is monsoon etc. Consider rules like if it is raining everyday then possibly the season is monsoon, or if it is monsoon then there is high chance that there will be rain today.

Now the condition is, if the season is monsoon and supposedly my fact is it is really the month of July which is the season of monsoon so the fact is true. Given this fact to be proved we can infer a new fact that possibly it will rain today that is another fact we are inferring but we are not stating any action as we had done if the right hand side were shut off the heater, open the tap etc. So the consequent of a rule can be either an action or can be a new fact. If it be an action that action is actually executed by the interpreter and if it is a fact then that fact is added to the fact base of the rule based system.

Also another important thing is that, along with the fact degrees of certainties can be associated. As we saw the rule, if the season is monsoon then it is possible that it will rain today. I can make the same statement in different ways. If the season is monsoon then there is 60% chance of raining today. I can also say if the season is monsoon then it is highly probable that it will rain today. Now it will rain today is not coming as it is but I am also adding some possibilities some certainties with this statement.

So when I infer a new fact what is the new fact that I am inferring?

Given that this season is monsoon my fact inferred will be possibly it will rain today, it is 60% probable that it will rain today. So it is possible that I can associate some sort of uncertainties with these rules. Now let us again think back at what happened to logic. In simple propositional logic or predicate logic we had an assertion which would either be evaluated to true or false. These were the two possibilities and there was nothing in between. But here along with these facts I can add on some certainties.

Later on in intelligent system we will see how we can handle such concepts. For the time being remember that in the case of rules or rule based system when we infer a fact this can be a hard fact which can be either true or false or it can be a fact which is associated with some sort of certainty, some degree of belief or disbelief whatever it is. So it is no longer a black and white decision there can be a grey region in between. Now we can also have different control strategies which are often heuristic. We often need to take decisions in real world where we do not have all the possible information we need.

Often we have got very partial information about scenario. Suppose you want to travel from a place A to place B you know some facts that there is a bus route from A to B and there is also railways connecting A to B, you do not know exactly what the fares are. But if you really need to save money then probably you will select bus travel because your other domain knowledge in general tells you that bus travel is often cheaper than train travel. It may not be always true but let us assume that there is a scenario as such. So you have got some partial information and based on this partial information and the other domain specific knowledge that you might have got through some other means or general knowledge that we have there are some thumb rules of making decisions.

And we often use those to take the decisions. And we can say probably bus will be a cheaper mode of travel so going from A to B for low cost bus would better way of going. Now these sorts of decisions we often take need not always have very strong logical bases but these are often very much practical and we often take such decisions. This particular scenario is also called as heuristic and such heuristics we will be employing for rule based systems and we will also see for different control schemes.

(Refer Slide Time 18:08)

**Example Rules**

- If it rains today the roads will be wet today  
Rains(today)  $\rightarrow$  Wet-road (today)  
If Rains(today) Then Wet-road(today)
- If Rains (today) and Not-covered (roads)  
Then Wet-road (today)

Which are facts?  
Facts are also generated

Here is an example of a rule based system. If it rains today roads will be wet. This is clearly understood when we write in English. We can write this in the form of logic as rains is a predicate, rains today implies implication, wet road today or in the form of a rule we can say if rains today then wet road today. You can see the correspondence between this logic statement and this rule. Again there is another rule I added here; if rains today and not covered roads that is if the roads are not covered then wet road today.

Now look at the difference between these two rules. This rule is obviously more specific because it says that the road will be wet today if it rains today and the roads are not covered. Whereas this is a statement which is much weaker if rains today then wet road today. Now look at the antecedent part of this rule; rains today is an antecedent and wet

road today is a consequent. In this case we have got a number of antecedents rains today and not covered roads these are two antecedents and these two together we call the antecedent field or simply we call them antecedent.

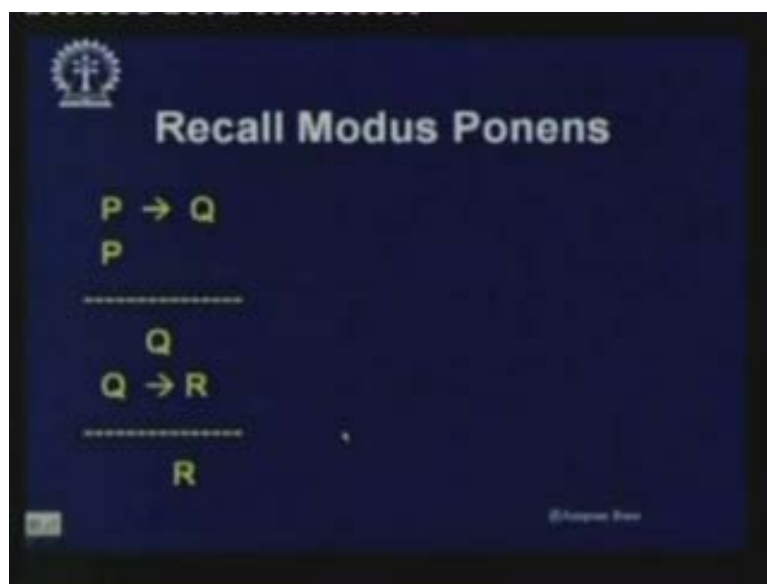
Now what happens to this rule if the fact is rains today, that means if rains today evaluates to true then we will conclude that the roads will be wet today, this part wet road today. On the other hand, when will we infer wet road today? We will infer wet road today if both these conjuncts of this antecedent evaluates to be true. In our discussion I will often refer base to be the antecedent part or the antecedent field and each of them are individual antecedents.

When does this rule will really work?

This rule will be active when both these antecedents since they are connected with an AND are both true. suppose the fact is, it rains today and the roads are covered today, according to this rule this part will be true but this part will not be true because the roads are covered then I cannot really infer that the road will be wet today. So in a rule we have got two parts one is the antecedent part another is the consequent part. The antecedent part can consist of one fact or can be a conjunct of facts or anti conditions.

Look at this consequent, there are two different types of consequents; some things can be actions and some thing can be assertion. Wet road today is simply an assertion that is adding to my collection of facts and since I know it rains today and not covered today I know one more fact that the road is wet today. Might be there is another rule, if it is wet road today then wear good shoes or something of this sort so this fact can be used for something more. Now which are the facts here? Facts are which are already known a priory but you can see that facts are also generated just as in this case in this example wet road today is a new fact that has been generated.

(Refer Slide Time 22:25)

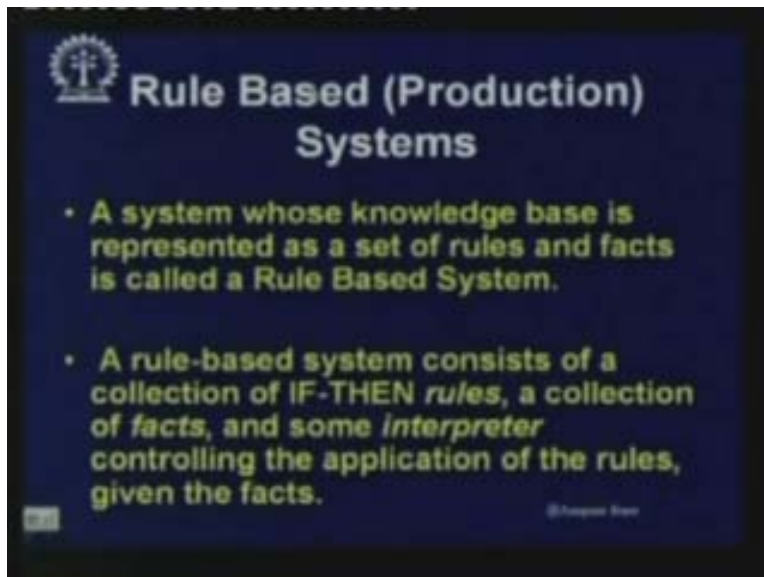




Now let us get back to Modus ponens which was a very common deduction mechanism we talked about when we were talking about logical inferencing. Here we just revisit Modus ponens. Here you see  $P$  implies  $Q$  and  $P$  so  $P$  implies  $Q$  is an implication and  $P$  is a fact that is given. So, given these two using Modus ponens what can we infer? We infer  $Q$ . that means if  $P$  is true then  $Q$  is true and since  $P$  is true we infer  $Q$  is true. Now suppose there is another rule;  $Q$  implies  $R$ . That means if  $Q$  is true then  $R$  is true. Now here from these two we have inferred  $Q$  and again from these two we can infer  $R$ . Therefore, if we consider this to be a rule and this to be a fact then we can infer a fact and using this fact and another rule which is the rule that has got this fact  $Q$  as its antecedent we can infer another fact  $R$ .

Therefore starting from this given fact  $P$  and these two rules  $P$  implies  $Q$  and  $Q$  implies  $R$  I can first infer  $Q$  and then  $R$ . So this rule is allowing to infer  $Q$  and inferring  $Q$  is allowing me to infer  $R$  so this phenomenon is also known as rule chaining where we are inferring a new fact using a rule and with the newly inferred fact I am applying another rule and inferring a new fact and the chain can go on. **So let us now come to rule based systems once again.**

(Refer Slide Time 24:33)



Rule based systems are also known as production systems. Those of you who have worked on compiler you have seen that often grammars are specified in the form called **Backus normal form** which is also a set of productions. Find the similarity between those statements which are given in Backus normal form and the type of rules that we will be discussing. Certainly there are quite a few similarities. A rule based system which is also known as production system is a system whose knowledge base is represented as a set of rules and facts. So we were talking about knowledge representation earlier. Logic was one form of knowledge representation.

We will see different other types of knowledge representation as well. Rule based system is therefore one type of knowledge representation where the knowledge is represented as a set of rules and a set of facts. We have already seen this, a rule based system consists of a collection of rules which are in if then form and a collection of facts. Now with these rules and facts we would not be able to infer anything unless there is another interpreter or control engine whatever name you give it inference mechanism which will look at these facts and see which rule is applicable now and apply that particular rule and generate a new fact because in my rule base I will have thousands of rules.

Now, given a particular objective a particular problem I want to solve all these rules may not be relevant. Therefore it is essential that I have to find out only those rules which are relevant at this particular point of application. So, that task is done by the interpreter. And in today's lecture we will see how the interpreter applies these rules, how the interpreter selects these rules and what does this interpreter really do to infer new facts and thereby solve some problems.

(Refer Slide Time 27:12)



The rules are represented in the following form; if antecedent then consequent, and antecedent can be a conjunction of number of antecedents. When the antecedent part is NULL it is also possible that I will just have a consequent. If the antecedent part is null then obviously it becomes a fact that is a true fact. In general we will assume that a fact is true. Now rule says that if these conditions are true then I can infer a new fact. But if no conditions are given then that consequent stands on its own merit or on its own right so it is true, it is a fact. Structures like while one, while two is always true. So, similarly if there is no condition attached to a rule then it becomes a mere fact.

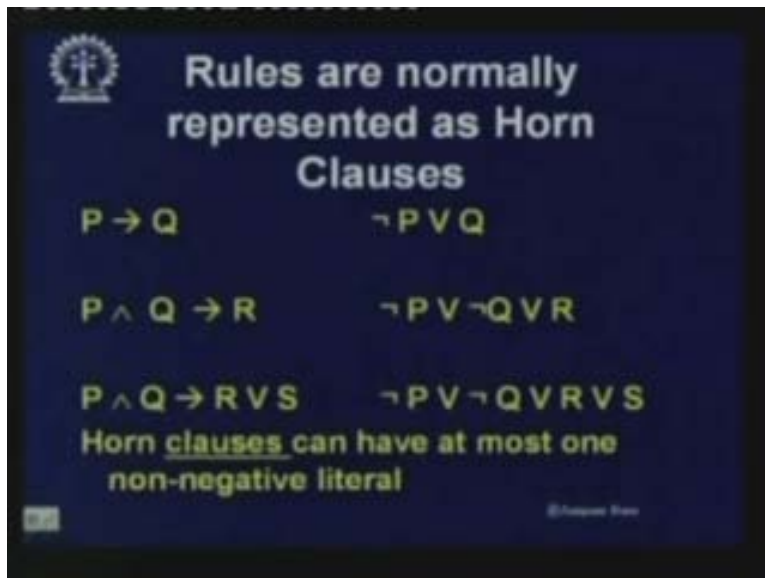
Here is a statement that I am making; rules are normally represented as horn clause. There is reason for this. Before being introduced to horn clause let us recall what a clause or a clausal form is. A clausal form consists of disjunction of literals. For example, p OR



NOT q OR r that is a clause or a particular form. And implications can also be converted to clausal form. Horn clause is a particular form of such clauses and it is referred that rule based systems based on horn clauses.

Let us look at this implication; P implies Q we will recollect the clausal form of this which is NOT P OR Q because this implication means if P is true then Q is true. Now if P is true then Q is true otherwise P is not true so NOT P OR Q is the clausal form. Similarly P AND Q implies R is a rule form. I can convert to the clausal form NOT P OR NOT Q OR R is a clausal form. Now look at this; P AND Q implies R OR S.

(Refer Slide Time 30:06)



If I convert to the clausal form it becomes NOT P OR NOT Q OR R OR S. This is not a horn clause, what is a horn clause? If we call anything a horn clause the first condition it must satisfy is that it should be a clause, there must be clauses. Horn clauses can have at most one non negative literal. Here Q is a non negative literal, here R is a non negative literal, there is only one non negative literal but in this case we have got two non negative literals so this is not a horn clause. So these are acceptable rules but this is not typically in a rule based system. It is not the case that you cannot write rules like this but usually it is not encouraged and the reason is there are some severe consequences to this.

If I use this particular form which is not a horn clause because it has got more than one non negative literal what is the problem? Suppose my fact base says that the person is rich that is P and the person is large hearted so P AND Q. Then the person will donate OR the person will build a school or whatever. So there are two possible implications we can take. Now, in the case of an implementation or execution the interpreter will really be in a problem to select which one of these consequents to take, should it infer both because I do not know whether both of them are true.

This part will be true if any one is true or should it infer anyone? If it has to infer anyone then which one should it infer?

Those problems really crop up but those problems are absent in these cases P implies Q or P AND Q implies R so we usually stick to horn clauses. That means in the consequent part we do not [ ] 32:49. Now there are two more terms which we should first understand. For a rule base system we have got a rule base and we have got a set of facts which we give in a fact base. Now these rules are evaluated to be true.

When is a rule applicable?

A rule is applicable when all its conditions evaluate to true. If any of the conditions evaluate to false then that is not applicable. Now it may be the situation that given a set of facts and the set of rules more than one rule can have their conditions satisfied. Therefore more than one rule is applicable. These rules are triggered and out of these triggered rules that means those rules have all been enabled to be fired. Now, when we apply a particular rule we say that that rule is fired.

What happens when a rule is fired?

When a rule is fired then the consequent part as to whatever is specified in the consequent is inferred or that action is taken. So, if that new fact is inferred then this fact is added to the fact base. If the rule just specifies some action and if that action is taken then that action creates some change in state. For example, as I was giving the example put off the heater then the heater is put off so the state changes wherein earlier the heater was on and I changed that fact I delete that fact heater is on and added a new fact heater is off. So, in either case there is a change in the fact base when a rule is fired.

But it is not the case that all the rules which are enabled are fired. I can either fire any one of them, if I have to fire any one of them or to make a decision as to which one should I fire so all these things are there. But first of all let us make a decision that a rule is triggered when all its antecedents evaluate to true and the rule is fired when the actions stated in the consequent part or the inference related to the consequent part is inferred or is taken.

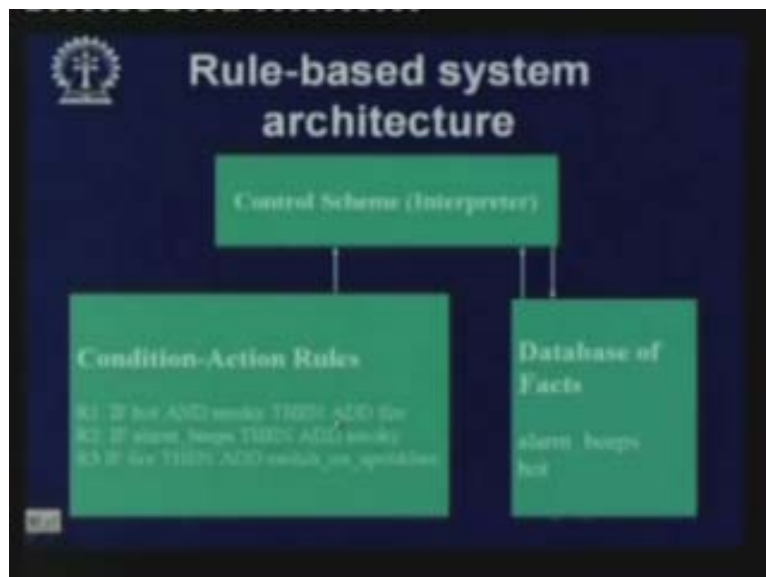
Therefore we will make a distinction between these two. Now look at this architecture. This is an overall architecture of a rule based system. You can see this big rectangle and you forget about the rules which are written over here. All these rules here constitute the rule base whereas we can have a number of facts stored and that is known as the fact base and here there is a procedural component which can have different names control scheme, interpreter, inference machine, control engine, etc different names which are given.

(Refer Slide Time 35:38)



Now look at this arrow, what does this control scheme do or interpreter do? It is looking at the facts and looking at the rules. And is trying to see given the particular set of facts which are the rules which are applicable which are triggered. Now, out of the rules which are triggered it will fire one or more than one and consequently when the rule is fired a new fact will be generated and that will update this fact base. So, at the top level we look at three components in a rule based system namely the rule base, the fact base and an inference machine. These are the three basic components of rule based system.

(Refer Slide Time 37:07)



Next let us look into these rules. Here the rule is, if hot and smoky then add fire. Rule two says, if alarm beeps then add smoky. Rule three says, if fire then add switch on sprinkler. So here we are talking of a toy problem where we have got rule based intelligent system installed which has got some sensors put on in the room and its sensing temperatures it is trying to see whether the temperature becomes very hot, and there are also smoke detectors so it is trying to see whether there is any fire and if it infers that there is fire it will automatically start on sprinkler or it will set up an alarm which will make people run and switch on the sprinklers. **Therefore it is that sort of a scenario we have here.** So we have a toy problem modeled with these three rules.

What is there in the fact base?

Alarm, beeps and hot so there are two facts. One is alarm beeps and hot. Since the alarm beeps what will this control scheme do? It will look at these rules. Now which rule is enabled? I can see also hot, so this rule has got this antecedent satisfied if hot, yes, true so this is satisfied. And smoky, but I do not know anything about smoky in my database of facts so I cannot say that it is smoky so this part is false. Hot is true but this smoky is false therefore this antecedent part of this rule is not completely satisfied. Therefore this rule does not apply. So this rule is not triggered and it comes to the second rule. If alarm beeps, and in the fact base alarm beeps is there so this part is true so this rule is obviously enabled because it has got no other antecedent. This entire antecedent field this antecedent part is satisfied.

The third rule is, if fire, now I do not know anything about fire here and nothing is here so this antecedent is also not true so this rule is also not triggered. In this case we have got only one rule that is triggered that is rule R2. So the control scheme then finds that only R2 is triggered so it has got no choice and it fires that. And what is the consequent part of this rule R2? The consequent part says, if alarm beeps then add smoky. So when it fires it immediately adds to the database the new fact smoky. And you can well guess what will happen next. Next the control scheme will again look at rules which are being enabled as fresh. Hot is true so this part is true and now smoky is also true. Since smoky is true this entire antecedent part is true. R2 is also true but R2 is already fired so we are taking up the strategy that I will not fire one rule time and again, it tries for the third rule fire but I have got nothing in the database as yet which tells that there is a fire. Therefore only this rule is enabled this rule is fired and what is the consequent of this rule? It is add fire, so a new fact is added to the fact base which says fire. So what am I inferring? I am inferring that there is a fire.

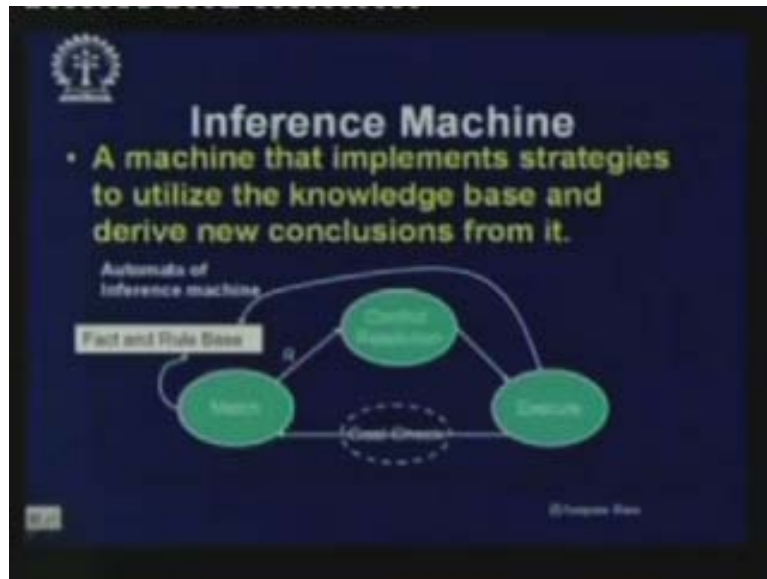
Now I again go back to this cycle and look at this third rule, the third rule is now enabled if fire then add switch on sprinkler. Now this third rule will be fired and sprinkler on will be put in the database. So essentially this is the way in which a rule based system will work. Now, if you have noted this example very clearly then you will realize that this can be modeled in the form of automata of inferencing.

What is that automata?

There are three phases, now what are the components that I have? I have got a set of rules, I have got a set of facts and the inference machine is also there. So what is

happening? First the inference machine is looking at the fact base and the rule base. I am trying to find a match **as to which of the rules are triggered**? If it finds that more than one rule is triggered then it will have to select some of the rules may be one rule or may be a subset of these rules. For the sake of simplicity for the time being let us assume that we are selecting only one rule and that rule is fired that is executed. And when that rule is executed a new fact is generated in the fact base. So we can essentially carry it on in the form of cycle.

(Refer Slide Time 43:50)



So here is what is depicted, the inference machine or the interpreter is a machine that implements the strategies to utilize the knowledge base and derive new conclusions from it. How it is being done? Here are the automata of an inference machine. There are three phases. First is the match phase then there is a conflict resolution phase and then there is a execute phase. Now in the match phase the inference machine is comparing the fact base and the rule base and it will find out some rules which are matching. If no rules are matching then we cannot proceed any further. But suppose there is a set of rules which match then those set of rules will be used by the next stage which is the conflict resolution stage.

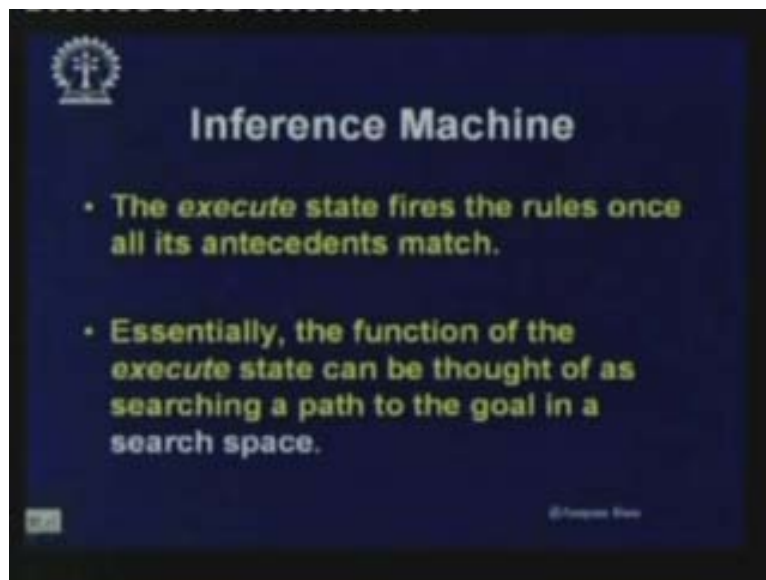
Where is the conflict?

The conflict is among the rules. There are five rules which are triggered in the match phase. Now which of these rules I shall be firing? For that I need to have some strategy which is often called the heuristic strategy. We often use heuristics but using heuristic what we do is the conflict resolution strategy. I will be selecting one rule or more than one rule. Of this subset of match rules which part of the subset I will be executing is decided in the conflict resolution phase. So we come to the conflict resolution phase and after the conflict resolution phase we decide on which rules to find. And then with that information we go to the next state which is known as the execute state.

In the execute state those rules are fired and when the rule is fired, here in the match phase we passed on only the triggered rules to the conflict resolution phase. Now in the conflict resolution state we have resolved the conflicts and we have selected some rules to be fired. When the rules are fired then the firing causes a change again in the new fact base the new fact scenario. After this execution you will again come to the match phase. But how long will it continue? We will come to the match phase and in the match phase again the matching will be done and we will see which rule is enabled again, we will do conflict resolution, we will execute and this cycle will continue. Now how long will this cycle continue? This cycle will continue until the goal or the objective with which we are trying to solve the problem. The problem we have, we are trying to solve that and that we call a goal, and as long as the goal is not met we should try to do this. But it may be that we have not been able to solve the goal and we do not have enough rules at our hand.

Rules are essentially representation of knowledge and knowledge is always limited. And often there are situations when we the human beings fail to solve a problem because of lack of knowledge may be at times. So, in those cases we try till we exhaust all the knowledge components we have so we will carry it on but if we solve a particular problem then we will stop otherwise we will exhaust our knowledge source and then stop. So this cycle will continue and after every execution phase it will go to the rule base and after **every execution** it will check for the goal whether the goal is met. If the goal is met then we will stop otherwise we will carry on. This is a very fundamental cycle of rule based system. Here is a summary; the execute state fires the rules once all its antecedents match and essentially the function of the execute state can be thought of as searching through a path through the goal in a search space. Here I have added a new name search space.

(Refer Slide Time 48:37)

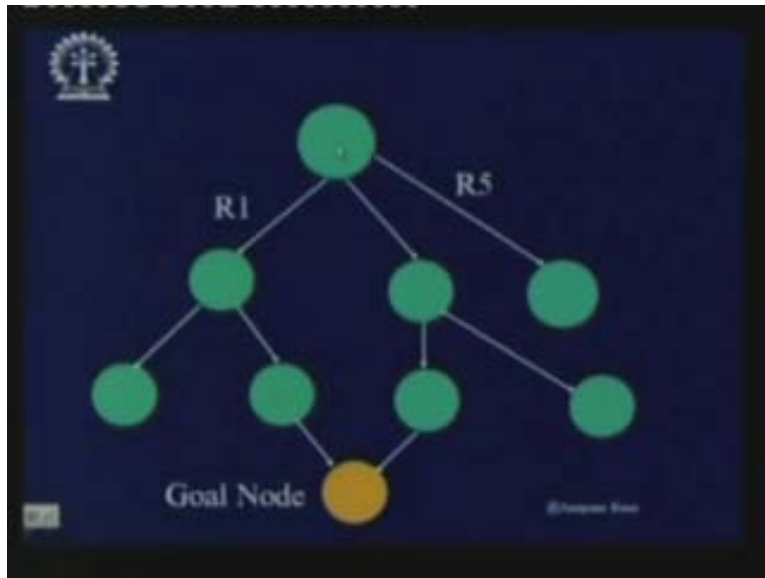


So what I am trying to take here is that, through these rules we are essentially trying to depict all the possible ways in which a problem can be solved. Our objective is to solve a



particular problem that is a goal and in order to do that there may be different alternative paths all the paths may not lead to the goal but there are paths which may lead me to the goal. So, effectively our task is to search through this entire space which I am calling the search space. Now here is a search space node. Typically a search space is denoted by these states. This state I am showing is the start state and I have got a goal state also here. Orange node is the goal state and the root node of this is the start state.

(Refer Slide Time 50:04)



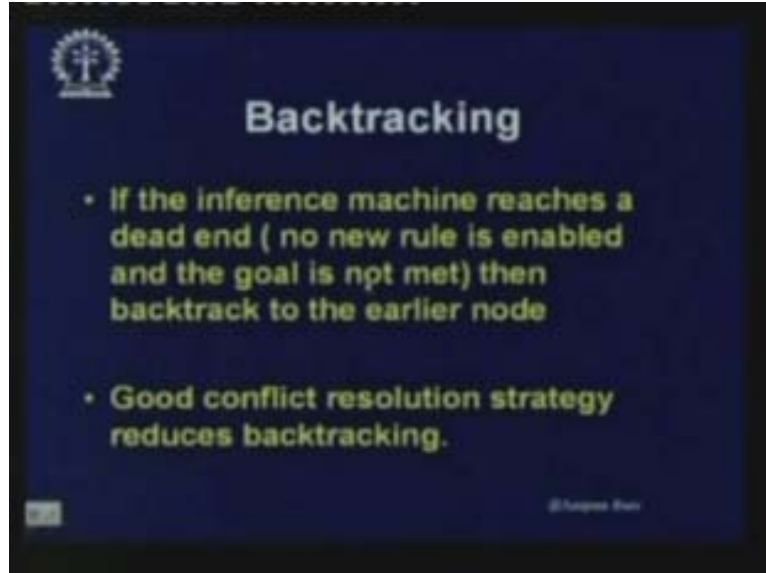
Now what do I have at the start state?

I have got my rules and I know about the goal and right now this goal is not solved so I start my rule based system here. And with the facts and the rules I have I try to see which of the rules are applicable which are triggered. Suppose at this point three rules are triggered, I have just named only two of them; one is R1 and the other is R5 but I am denoting these set of rules using this tree structure. Now what this state shows is that the result of my match as shown here that has resulted into three rules which have been passed to this conflict resolution module. And in that suppose I have resolved in favor of R1 so I can come to R1 and when I fire a particular rule R1 I will be adding new facts to the rule so I continuously refer to this diagram (Refer Slide Time 43:50).

So I have resolved in favor of R1 and they have been executed and new facts have been added to the fact base and that might have enabled me with two other rules and I can select any one out of them. Suppose I have selected this one and fired this rule so here again there was a conflict. At this state again a match has been done and two rules have succeeded and these are triggered given this new state. And I have fired this rule and I have come to this new state. And as I come to this new state and if I fire this rule I will come to the goal node. But what are my possibilities? My possibilities are that I could have followed this path, I could have followed this path, this path or might be at this conflict resolution strategy I could have followed another path. Suppose at this point after firing R1 I come to a new state and I fire a new rule and I fire this rule and come to a



(Refer Slide Time 54:46)



Backtracking is a very important methodology of exploring the search space because it is often not possible to really make the correct choice all the time it may not be always possible so we will select the rules which are not proper given a particular context. And obviously conflict resolution strategy plays a crucial role over here if we have a good conflict resolution strategy then obviously the backtracking will be reduced. If you have good conflict resolution strategy then we will be reducing the backtracking.

Now, in this lecture we conclude at this point where I have just tried to give you an overview of what rule based systems are and how they work. In the next lecture we will look at other aspects, we will look into more examples; we will look into more details of inference mechanism. But I hope that you understood how a set of rules really help us in solving the problem. You should remember a couple of facts namely the distinction between rules and facts, the distinction between match phase conflict resolution strategy and execution strategy. You must understand what is an inference mechanism? An inference mechanism actually selects the rules to be fired given a set of facts. So given all these we will move on to the next lecture where we will be going into more details of rule based systems.