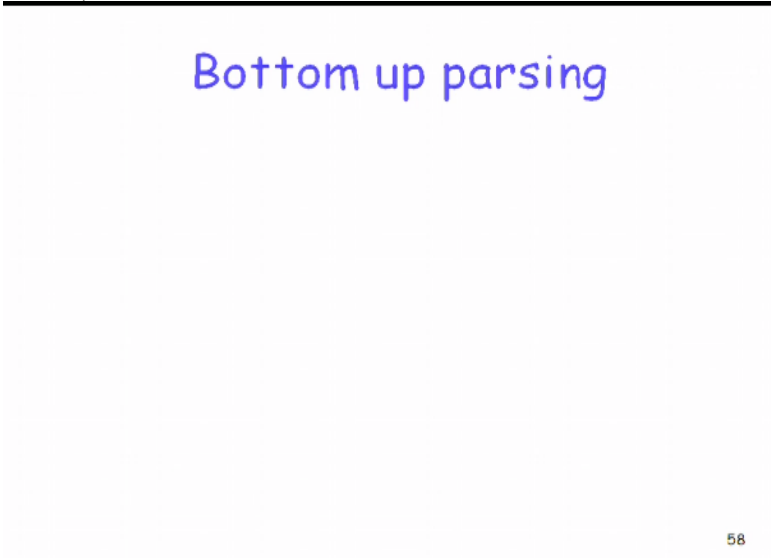
Indian Institute of Technology
Kanpur
NP-TEL
National Programme
on
Technology Enhanced Learning
Course Title
Compiler Design
Lecture-09

by…
Prof. S.K. Aggarwal.
Dept. of Computer Science and Engineering.

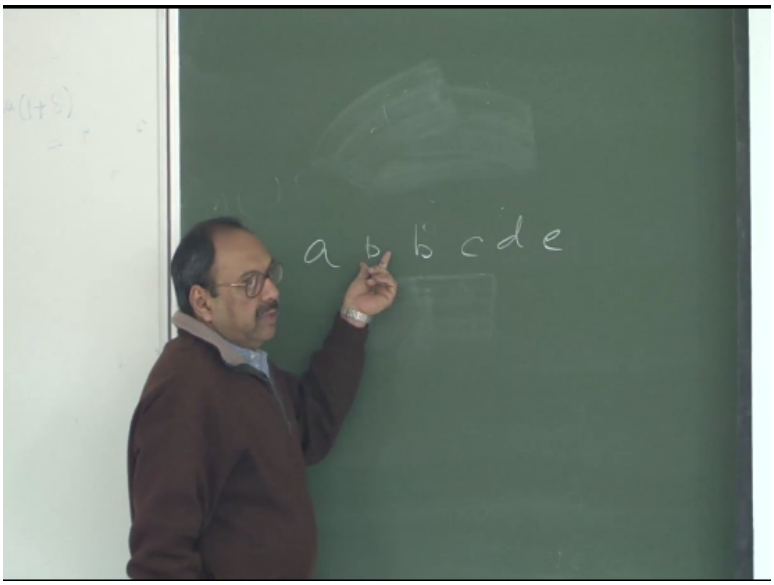
(Refer Slide Time: 00:17)



Good morning so let us start on new topic called weather and on finishing the bottom up parsing so bottom up parsing is a method where you start constructing the parse tree of the leaf nodes and then try to reduce the whole of the string with the parse symbol and if we succeed in that let me say that the string belongs to language is specified by the language by the down so in this case you want to construct the parse tree from the input that the leaf nodes or another way to look at it is that if I say that W is a valid string in the language which is specified then I should be able to somehow reduce w to start symbol s this is our way of saying.

So let us look at the grammar and so let us take this grammar and to see whether this thing is specified by the and what we want to do so here is the grammar where we have a start symbol s and then corresponding to the production of non terminal a I have two rules and then I have one more rule and then there is the string I want to parse.

(Refer Slide Time: 01:43)

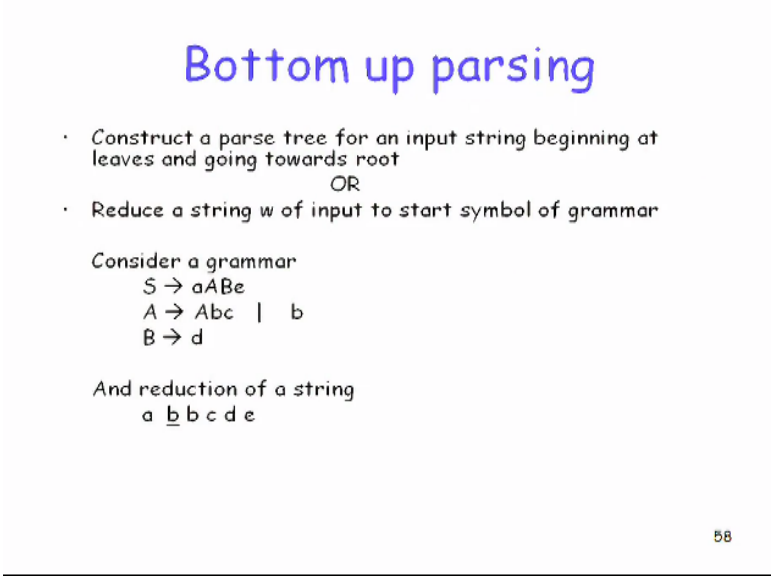


So let us take this grammar and try to see whether this spring belongs to the language which is specified by the language and how do we go about and what we want to do is we want reduce the string to the start symbol so what we try to do here is find out is we try to find out the substring here it matches the right hand side of a production and I a m going to replace by its left hand side and we continue and hope this I valid string in the language specified by this grammar then you will be able to reduce the whole string eventually to S right.

So the first thing I do is that here is b which matches right hand side of a production which says A goes to B and I replace this by A and after reduction my string becomes other become sentence form now again I look to this and find there is another B o I just go ahead and try to replace that by its left hand side.

And when I do that then I get a string like this now let us again try and see if that I can find a pattern on the right hand side so b can match d this right hand side then if I replace that by b what will happen so I m replacing this by B I reach here now how do I proceed from this point can you find a pattern on the right hand side which matches one of the productions no so it appears that the choices I made took me to a situation where we took the string actually does not belong to actually this string language which is specified by the language.

(Refer Slide Time: 03:59)



Now I request everyone that from next class onwards either you reach here on time you know that there is class at night and there is no reason mean and you know that you can reach here five minutes early I mean it disturbs the class we discuss we have a pretty good class of which we have already spent seven minutes and we class keep coming in and I am sure I'm in the next five ten minutes when you know that there is class at nine now I can't understand why you cannot come in time I mean if you wake up at 6:30 you wake up at 6:00 wake up five minutes before that you will reach here at night I dont say any problem this habitual habit of saying that there is a class at nine time I can always and the true then I can spend and it is just not an acceptable situation so next class almost make sure you are all on of time .

So when we started making these choices so at some point of time suppose I make slight different choices then let us see what would have happened so when I was so another than going in this direction let me just keep this here and try to my rest of the strings on this side and when I say that this reduction happens okay I am going to this situation and now when I make this choice I was saying that match this B as the right hand side instead of that I make a different choice and I say make this match this ABC which is right hand side of a production and then replace it by itself .

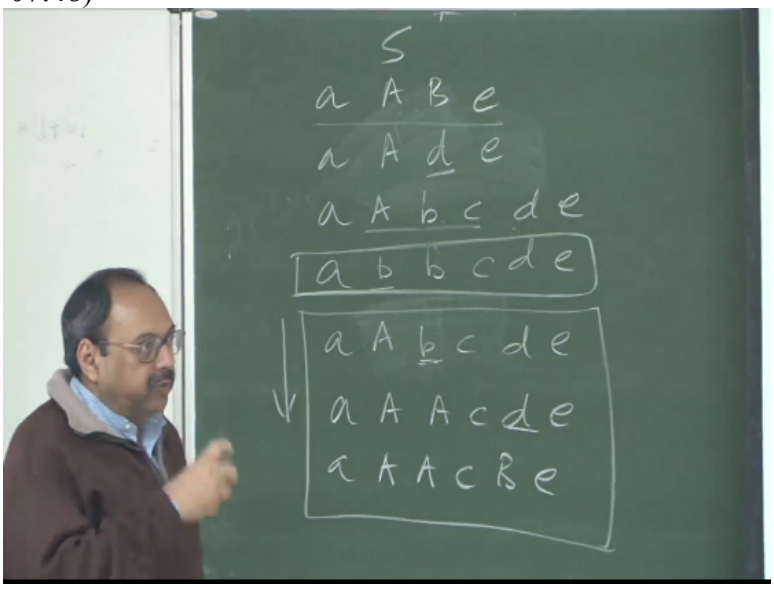
So if I do that like this and now again if I look at right hand inside pattern here which is B which matches right hand side of a production like this gives me. I get a screen like this and this matches now right inside of the first production the whole thing and here you to start s and therefore I can conclude that the string I started with this a valid string in the language I can

show it on the string all these steps but only problem which happened in this part was at some point of time rather than making this as a choice of matching.

I made this as a choice of the solution so the same thing like if you recall because saying at some point of time of parsing which terminal to use here the issue is going to be which substring to match and if I have more than one choices and somehow I need to find a strong method of doing this kind of reduction which will always identify that this is the right string and this is not the right string to match.

So this is what will be going to the part of building the bottom of parser that when we process this grammar somehow this information will be captured so is this point clear that two different choices can lead to a situation which are entirely different from each other so this is what is captured here okay but let me also simultaneously observe that if I look at this complete reduction .

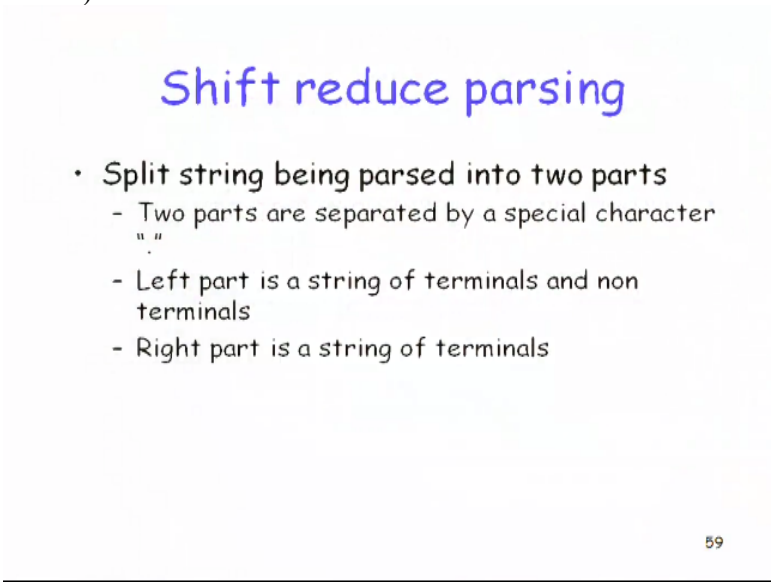
(Refer Slide Time: 07:48)



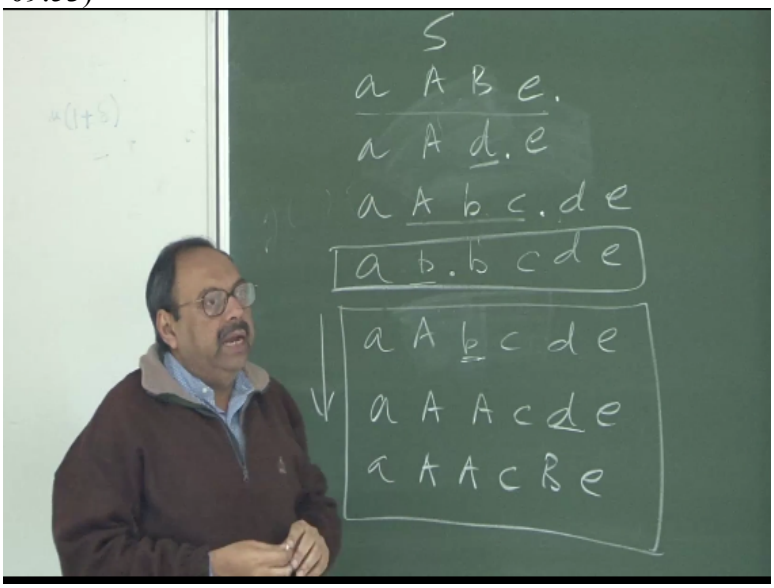
And I also look at the rightmost derivation of the string starting from s okay then I actually I am doing a derivation which is that most but rightmost that most which is talk down and it gives me the same string and now if you see the steps here it this actually is the rightmost derivation that means what I am doing is I am going in reverse of now I won't go into the proof of saying that if we are building a bottom of parser then the reduction has to be in the reverse of this information somehow will be captured .

When we start constructing the parser for grammar lines when we start the bottom of parsers are really the reverse of now these are also known as shift reduce parser now shift reduce somehow means you can get this feeling that when you need to use the stack but basically what that means

is that I can take any string and when I say that I am taking a string like this and I want to do a reduction concept to me now you can see that some parts I have seen and once I have seen some part than I am doing a reduction okay .
So let us put some imaginary symbol in middle of a screen and let me call it dot and then we say that two parts which are separated by the special symbol dot.
(Refer Slide Time: 09:29)



when you say that two dots which are separated by the special symbol dot when you say that because I am doing now a reverse of right post derivations then the left-hand side of dot is going to be a string of terminals and non-terminals and the right part of what is on the right of the dot is always going to be a string of terminals okay so look here what is happening okay.
(Refer Slide Time: 09:53)

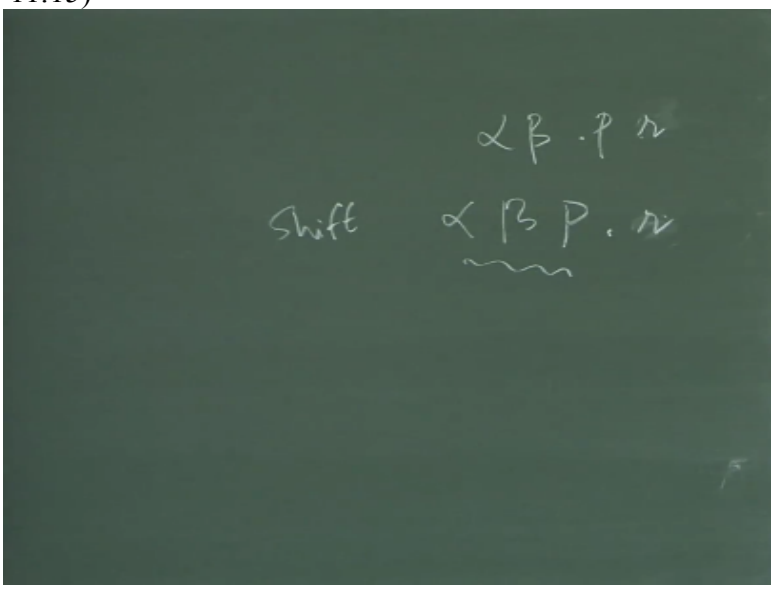


If I say this is the derivation I am doing then first we write this then from this B I derived this then from A I derived this and then from this A I derived this okay .So at any point of time what was happening that was saying that I will be somewhere here and then I said I am going to replace this then my dot must have been what I was just reducing here then this dot must have been here and then this dot must have been here okay.

Now you might somehow this dot appeared in these positions in my string then what was going on what was going on was that if you now see the right-hand side of the dot you only have either a null which is the final state and you are going to have only a string of terminals here but what you have on the left hand side is a string of terminals and non-terminals so basically this is my start state is that I will always start when I say that I am beginning this is where I started with that to begin with my dot will be here .

And then I say I will keep on shifting this dot along the string let me pick up another example and keep this example I use so that we can do this comparison but basically what we are doing here is that at any point of time if I say that this is the kind of computation .

(Refer Slide Time: 11:15)



I have something like this so on this configuration is telling us I have some string of terminals and non-terminals on left and what I have on the right hand side up basically and now when I say that I have something called a shift action shift action says that the symbol which is immediately to the right of dot must be shifted to the left of top okay so my configuration after this is going to be something like this or if you are using w with the whole string .

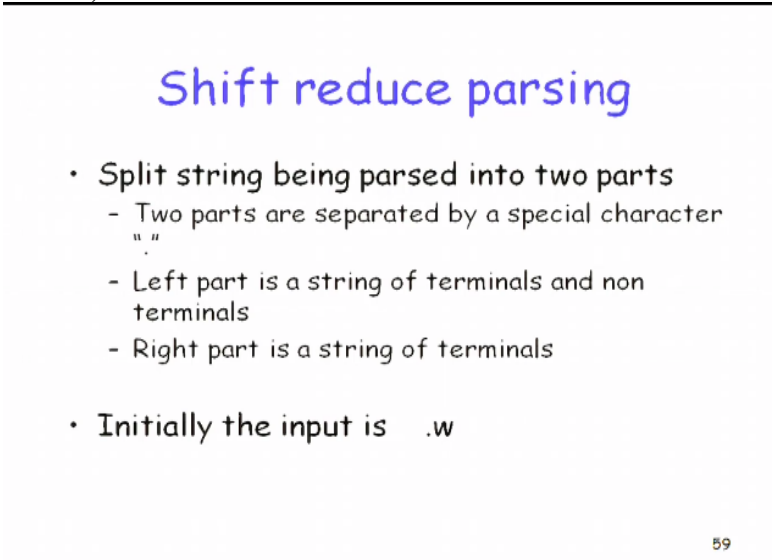
Let me just replace it with r my configuration will be something like this and what will be the new section new section is saying now see the example there that if you see something which is

immediately on the left of dot and which is matching the right hand side of production then we say that we are going to do reduce action ,so reduce section is saying that if this part of this starting from here matrix right hand side of the production match that and replace it by the left hand side of the production .

Now let me also alert that when you talk of bottom of parser is not as that you were starting from start symbol and by having to look ahead of one you were able to figure out production use whatever parser to begin with these slightly no need to be but as you get into it will start getting good so on this part that somehow you will find that this imaginary position of dot is going to create lot of confusion did we really give the definitions so initially my footage is that dot I to the left hand side of the string.

I am trying to match and what is the final position final position is going to be that on the right hand side of dot is going to be null and on the left hand side of dot is going to be start symbol.

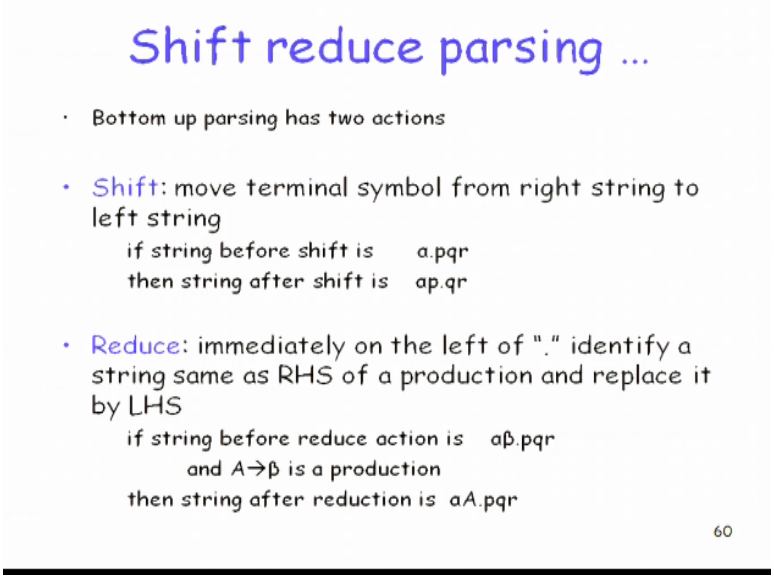
(Refer Slide Time: 13:24)



So my final position is going to be something like this which is here and initial position is going to be and then I have to see that what are the situations under which I do a shift and what are the situations under which I reduce. So for example I was in this case I said that do a reduce take it to B but when I shift the dot here that I am saying do not do a reduce here but keep on shifting and then do a reduce okay .

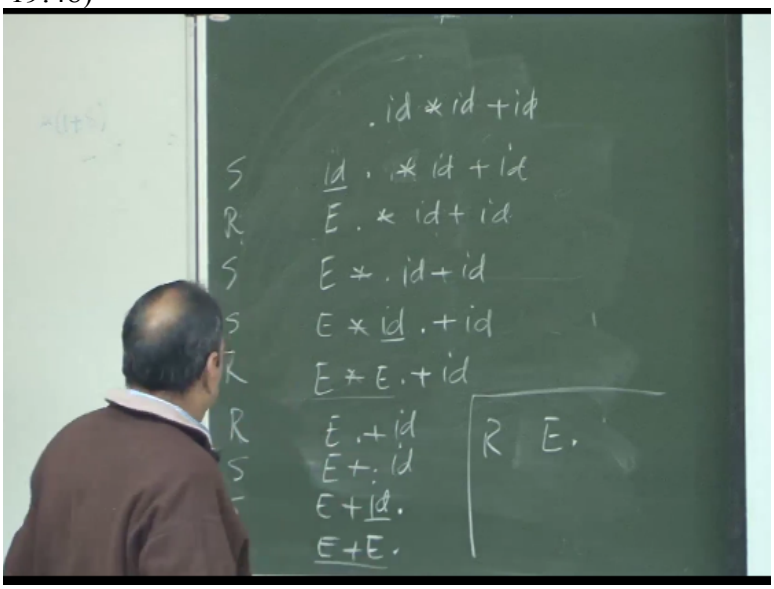
So you have a choice between doing either the shift or doing a reduce at least in this size of B and then I said that whatever is on the left hand immediately left-hand side and I reduce that to either I so raise question is that if I have a choice and the choice is saying that I do is shift or reduce in one of the choices. let me come conclusion saying string does not belong to the

language specified but this choice makes sense because I could reach my start symbol my parsing method should make sure the right choices right now there is some manufacturing like we did in parsing.
I concluded all these pin down that terminal expansion and look at symbol then use this similarly here we are saying that if dot is in certain position that means I have to do a shift or reduce and if I have to do a reduce than which move like to reduction if I can answer back then I should be eventually we will have so but right now have a positive once I create a possible take action only one action shifting I mean reduction is one action dot will remain here right so what I have done is I found a pattern which is on the immediately left of dot .
I found that pattern that is right inside of production and I just replace that by each step concerned similarly when I found that on the left hand side of dot there is a pattern which matches a particular production then we just replace that and I kept the dot here what I am NOT shown in between is that from this I must have done this shift .So let me take an example to reduce and then we come back to this answer okay.
(Refer Slide Time: 19:13)



So shift reduce parsing is it will have two actions either I can do a shift which says move terminal symbol which is on the right hand side of the left hand side of dot which this is my configuration which says alpha dot P Q R then after shift this is going to the alpha P dot Q R so P has been shifted on and then I have a reduce action which says that if I match something which is on the left hand side of dot and this production then.

I am going to reduce so if this is the pattern I have it says alpha dot P Q R production then and the production is A going to beta up then my configuration will be alpha A dot P Q R this is reduced action so b in this case I have shifted P to the left of R these are the equations.
So I keep on doing these mix of shift and reduce till I reach either the start symbol or error symbol start symbol mean that I am in the accept state saying that this is a valid string and error state saying that this is not a valid string these are the only two confirmations so let us look at this example and try to parse this particular string so again I just try to work it out for you so that please get a feel for what is going on and.
(Refer Slide Time: 19:48)

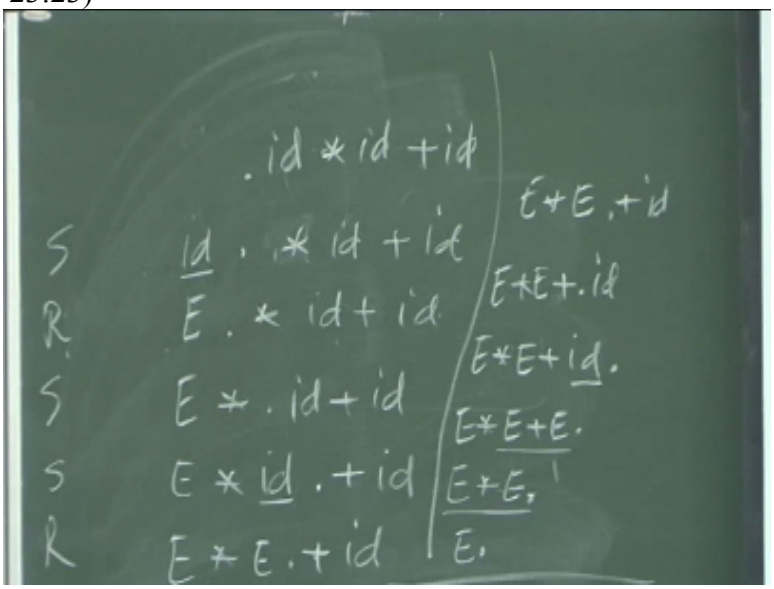


What we have is trying to say that I have I D *I d + ID this is my start string and my dot is and the grammar is on the top it says he goes so what I do now is first I will do a shift so my action is shift that it now becomes my string becomes of this complication and then I do reduce because I find that immediately on the left of dot there is a pattern which message is right inside of a production so my reduce is by action saying that I D gets converted into E and my configuration remains same.
Now once again I do a shift action when I say that E * . ID and then Id is configuration like this and then I do one more shift and this shift take me to a configuration like this and then I do reduce action which says that so let me underline whatever when I do a reduce section so here is an ID before I went to reduce section there is one ID which I will now do using a new section and then now I have a choice whether I want to shift or I say that E*e matrix right hand side out for production and right .

So what I decided at this point of time is that lets me do reduce action I am not telling you why I am making these dot while reducing instead of shift but let us assume that this is the action I think then what happens is this becomes my configuration and then I shift once again and shift action gives me first string like this and then I once again shift and this shift gives me this configuration and then I decide that I want to do a reduce okay.

And if I do a reduce this gives me and then I want to do one more reduce action matching this which gives me now and each my start symbol dot is immediately on the right-hand side so I say this is a valid string which identify true now you can also see in this case suppose when I have this choice of saying that rather than doing a reduce here I need a shift what would that happen would I have reached this state finally. So let us try that what was the second alternative in this case when I was doing string.

(Refer Slide Time: 23:23)



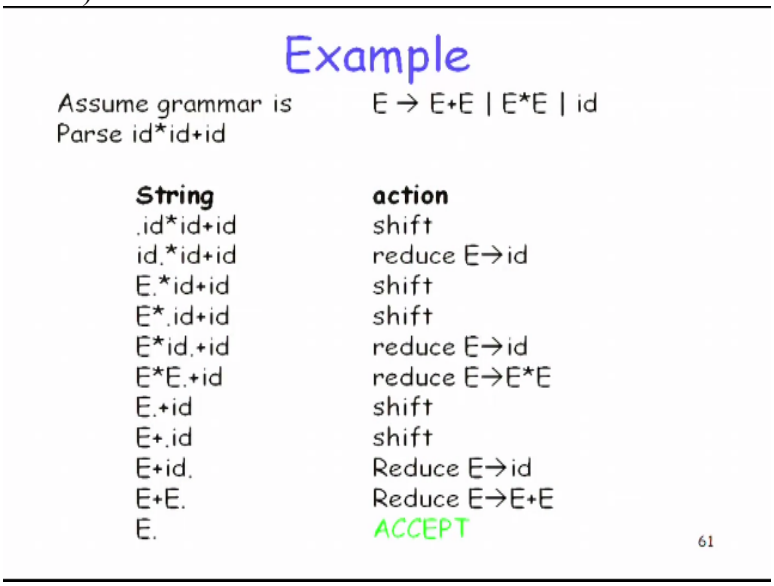
Let me say the time in this table in which is e * e .+ id and rather than doing the reduction I do a shift what will happen I get something like this I do one more shift and that makes me to this state now I do one reduce and deduction is now on this so this gives me this configuration then I o one reduce it says I match this as the right-hand side and remember that naturally I always wanted to do immediately starting on the dot position.

So I will not find a string like e*e and I say I will match that but I always find a string which is immediately on the starting from left so this will give me this reduction and then if I now reduce you almost have an idea perhaps that why this would have led to the wrong things and this is the right set of steps I do so is the second you find that if I am trying to if I know that this is

multiplication and this is addition then this has higher precedence and I want to make sure that this gets evaluated first okay.

Before this and therefore if I follow this sequence then I am doing things in this sequence although I somehow need to take care of all my business to set associative also might specify my grammars and so here is what we were just gone through that finally I reach this it off step I say this is except I am just doing shift actions okay.

(Refer Slide Time: 25:58)



Now when I start with this string when to do a shift reduce and then and my whole algorithm is going to be based on what equivalent to if I try to look at bottom down parser equal to saying the first and follow said computation I do some reprocessing here I do some similar configurations and try to take X number and encode it into parse table so that I can always is it beginning to sink him out and this tomorrow here when we started so continuing on shift reduce parsing .

Now let us also start simultaneously start talking about the implementation so whatever is on the left of dot I can keep that of this side and whatever is on the right of dot that can be string which is so this is what we do and the most important issue we want to raise here is to shift so reduce action should be only done when we know that and this is here now let us start thinking about the first example .

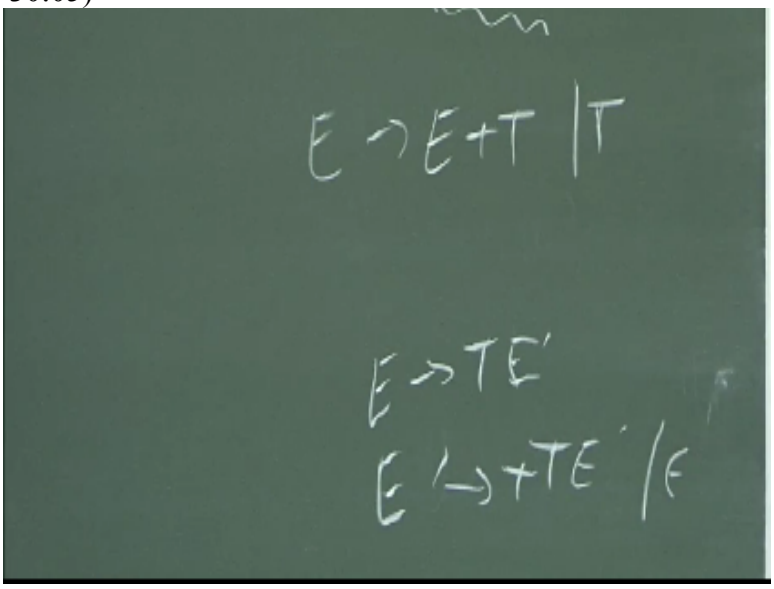
Let us do a block saying I cannot proceed and I input conclusion saying at this is the wrong string but if I it is the right set of reduce actions when I was able to reach the left hand side A which is the start symbol so therefore we are saying that the reduce action I should take only if the reduction is going to take me somehow to the start symbol if it is a valid string the language

does not matter so this is somehow is this is a fact which needs to be captured somehow when I start creating my parse table .

So let us look at some of the properties of bottom up parser before we go any further one fact this is more powerful parsing technique as compared to the top-down parsing and this lots of numbers is known as alum numbers it is more expensive than a level-1 or equivalent and when I say expensive expensive in terms of creating the parse table to put in more effort create the parse table also is going to be large it is will to occupy more space as far as parsing itself is concerned that is going to be still the linear time and I will not affect me okay.

So it is more expensive than a little one and therefore from later point of view you can see that it can not only handle left recursive grammars but it can handle almost all programming languages and therefore this also remains the method of choice for creating parsers for most programming languages and also it is a natural expression to write in the programming syntax .

(Refer Slide Time: 30:05)



So for example when I had these number like e e +t or T my top down parser was not able to handle that and I had to rewrite my grammars into this form and so let me just write this form if you think this grammar it is not really natural to figure out that what we are talking about - part of the expression class if I read this I can immediately figure out what I am talking about so this is a much more natural way of expressing language than this kind of notation which can handle parsers .

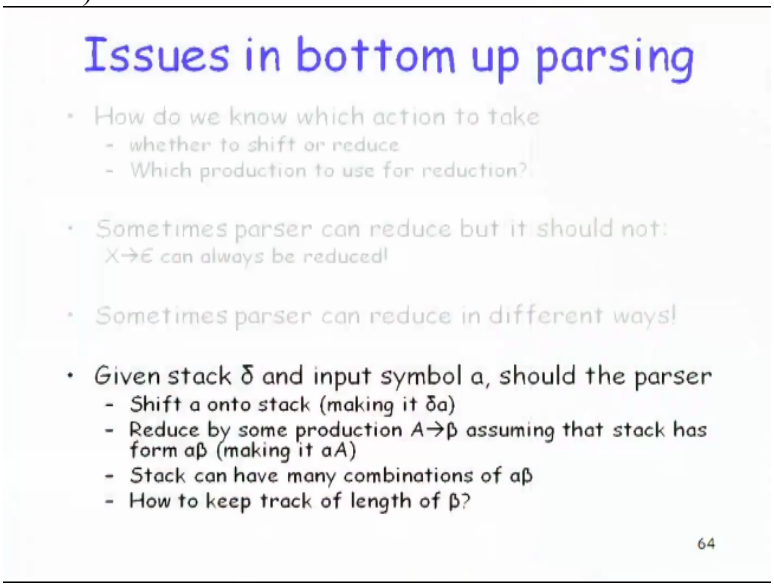
The act is a prime example because the first rule of parser also another interesting thing is that when it comes to vertical v the bottom of parser guarantee that it will never do a wrong shift they pass the bottom parsers which can either be counted rather than saying these start certain

symbols in this case we immediately be able to pin that down and therefore properties but as you will find that we be able to then look at these properties and these properties will just stand out as we construct the parser .

Now let us also simultaneously look at one of the issues that we face in a bottom-up parser so the first question which you have not answered is that this when to shift and when to reduce it so in this case I was doing reduce and in this case I was doing a shift I should be able to clearly answer that and then I should also be able to say that which production I want to use for reduction because there may be situations where I may have more than one right hand side matching production okay so sometimes I can have this reduction which says expose to a challenge and always be done okay.

But we do not want to do that so this question needs to be answered very clearly and sometimes parser can reduce in different ways like here it can reduce in different ways so what we have to remember here is that if I take stack so look at it this way so let me just explain this point because when I am doing let us keep this example. I this is it has lot of symbols .

(Refer Slide Time: 32:45)



Now this is my stack it has lot of symbol when I say that I am matching the right hand side of the production so this is so much that I can say this matches the right hand side of production and that means somewhere I say that if my stack is containing some string Delta which is a string of terminals and non-terminals I got to divide this into two part okay and this really is in one case this is the division in another case this I beginning needs to answer.

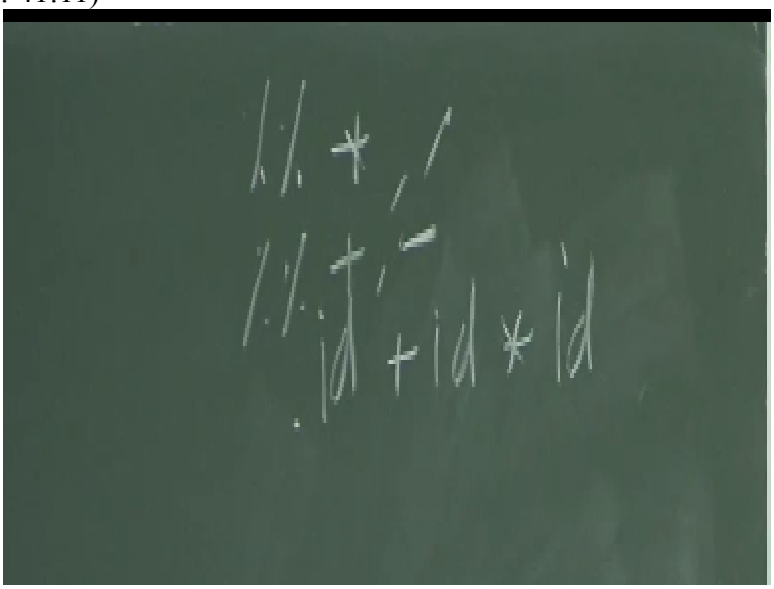
So this basically saying that suppose I say that Delta is my stack configuration and input symbol is a then shift obviously will give me Delta yeah but when I say reduce reduction can happen that

I can either take this part which matches the right hand side of the production or I can take this part which matches the right hand side of the production and this my answer will have to figure so what we start doing now is that how do we keep track of what is the length of this .
So if I say that I can have a combination of alpha and beta and multiple combinations then how will you give the length of this whether this is the right size or this is the right size so we will start giving some definitions so that at least we start building the parser so after raising all these issues let me give the first definition and that is it handle is something which has two properties so handle this right hand side of approach if I now look at this .
So what I was really doing was I was matching this ID I was matching this ID then I was matching e*e to handle something a string that matters right hand side of a production and the second part which is more important is that if I replace this right hand side of a production that is going to give me a spec within the reverse of right most derivation so unless these two conditions are satisfied .
Something cannot be designated so this was precisely the problem if you recall that first situation I had first situation was I started with this and I said this matches and I replaced this by its left hand side and then I alone is said I am going to replace this by its left-hand side so it matches like inside of a pattern but if I look at the reverse of rightmost derivation then this replacement was not giving me the reverse of right most derivation.
And therefore this was the wrong choice and therefore now if I want to define my shift use partial in terms of handle I can say even identifying handles and keep on replacing handle by its reference if I keep on doing that because I am doing now what is the handle now this is saying it is going to give me a step in the reverse of right most derivation so eventually lead me to the start symbol in some of this information will have to be captured in the method that each time I do a reduction somehow that is going to lead to a reverse step in the right most derivation .
So let us look at this so if this is string what will I do is I will start from here and shift this and then I will do a reduce then I will shift this then I will shift this I will do a reduce and then I you either a shift continue to shift this and then shift this will reduce then replace this and replace this without and you could so I have the same problem so you tell me which one of them is wrong .
So that mean that this grammar is not in a form where I can generate string without any problem, so what I the problem when I had more left most derivation o bottom of parsers is that I do not have ambiguous grammar actually it is an I want to make sure that the grammar is in form and therefore I wrote e goes to e+ t are saying that this has higher precedence than this and therefore when I had this choice between either doing a shift or reduce then I make the right choice.

And say that do not do a reduce the shift first and then no reduction so the problem for something like this suppose my configuration at some point of time was this would have in the configuration now I have a choice saying that either I do a reduce or I do shift and but because I define somewhere that this has higher precedence than this would have resolved in favor of a shift and a shift would have left to situation like this .

And then if I do a reduction here it Is a right reduction so you do not give precedence in the grammar you use some tool to give a business otherwise I said the matter should be rewrite the grammar so that you do not have ambiguous any it will always be and therefore I am writing my grammars all the times which is in this form right this grammar is not ambiguous this will always give you one left most or one right most derivation .But if I write grammar like this and I cannot write it but if I use a tool.

(Refer Slide Time: 41:11)



Then I will be able to use something like this it said that these have same associate and this has higher precedence than this therefore best method I said was like grammars which are ambiguous and then the first thing we talked about when we discuss now because no it was not in the reverse most derivation I showed you the right most derivation should I go back there okay let us go back to that point once again.

So this is the rightmost derivation and if I just start doing reverse most derivation if I reduce this P to a then that will not give me a stack here but if I do this reduction that is okay right in fact it was not and this is the derivation so I am saying that if I remember this fact that whenever I do a reduction action or I do shift action that should lead me to the reverse of right most derivation does not mean that I first right most derivation and check it.

I can always devise a method where I can take this grammar and when I convert this into tabular form this fact is captured there actually how do we write right most derivation because I have not shown you how to construct the parse stable this fact somehow will be captured in the parse table that what is the right hand see ultimately what I have to do is so let me go back to the last point what I was showing.

(Refer Slide Time: 44:30)

So here we are saying that a string that matches the right hand side of a production and whose replacement gives a step in the reverse right most derivation that means if I can have a method which I capturing this information yes you are going into a territory which you have neither business for life for which we do not even have tools so it is not symmetry first I mean look at it this way I defined the concept of a look at its symbol and I was reading it from left to right yes . Can I say the same thing about reading from right to left and say that rather than look at symbol and I have a tale simple no so symmetry does not apply so let us stick to the background and not stay that icons put dot anywhere are between dynamically replaced and if I start from there all right then huh then you can say that why go through this parsing method I am talking about a specific parsing method there are other parsing matrix which is not even worried about so natural language do not have such clean interpretation of the precedence and associatively then you have .

So let us do one thing okay let us move slightly ahead and then keep coming back till this issue gets resolved because if you get stuck at a point we may not be able to put at this point of time if it is not here at least by taking an example what are the parsing we know always is an issue when

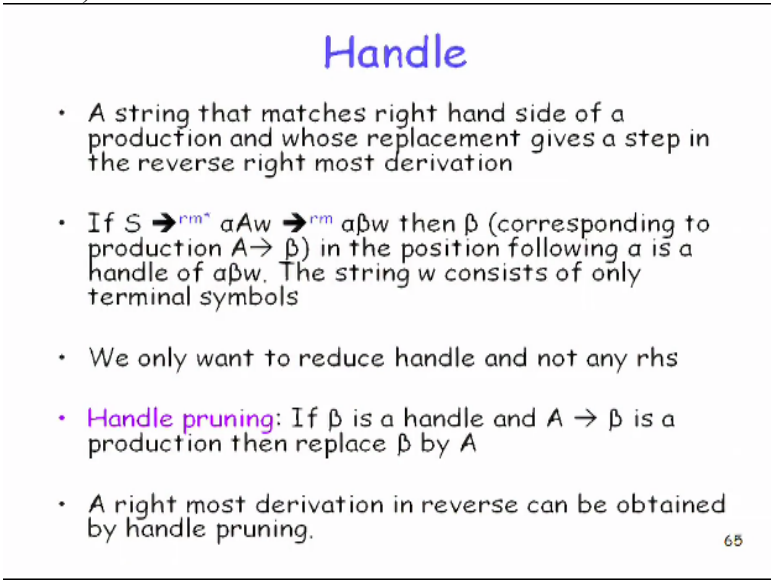
it takes time to sink in but once it inks in find with it okay so let us look at what a handle is and what we are saying is that if I staged .

I am doing a rightmost derivation once I handle means is that if s in a step of rightmost derivation takes me to a configuration which is α β w and then one step of rightmost derivation says that since w the string of only terminals in is the first non terminal on the right hand side if I see from the right hand side and there is a production which says a goes to beta then alpha aw gives me alpha P alpha beta that B so this is the rightmost derivation and the reverse of rightmost derivation will be that I look at beta and replace it by E .

So in this case we say that beta is not handle, beta is the handle only if such as that is possible to static post-derivation so for example here b is this piece the handle but this B is not a handle so it is not that something matches right-hand side becomes a handle but it also ful fill this property that it should give me this step so let us take a break today I will just finish this yes so what we want to do is we want to reduce only handle and not any time inside okay .

And then what is handle tuning handle tuning is saying that if we tie the handles and a going to beta is a production then replace beta by a and rightmost derivation in the reverse can be obtained by just by handle.

(Refer Slide Time: 49:55)



So the question now is in a bottom of parser how to identify them because if I can identify handle then my job is done okay so whole construction of whatever parser is through handle booming and by the pattern of that is something we will start discussing tomorrow and by this week end we will be able to finish at least one version of bottom up parsing.

Acknowledgment

Ministry of Human Resources & Development

Prof. Phalguni Gupta
Co-ordinator, NPTEL IIT Kanpur
Satyaki Roy
Co Co-ordinator, NPTEL IIT Kanpur
Camera
Ram Chandra
Dilip Tripathi
Padam Shukla
Manoj Shrivastava
Sanjay Mishtra
Editing
Ashish Singh
Badal Pradhan
Tapobrata Das
Shuubham Rawat
Shikha Gupta
Pradeep Kumar
K.K Mishra
Jai Singh
Sweety Kanaujia
Aradhana Singh
Sweta
Preeti Sachan
Ashutosh Gairola
Dilip Katiyar
Ashutosh Kumar
Light& Sound
Sharwan
Hari Ram
Production Crew
Bhadra Rao
Puneet Kumar Bajpai
Priyanka Singh
Office
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari
Saurabh Shukla
Direction
Sanjay Pal
Production Manager
Bharat Lal
an IIT Kanpur Production

@Copyright reserved