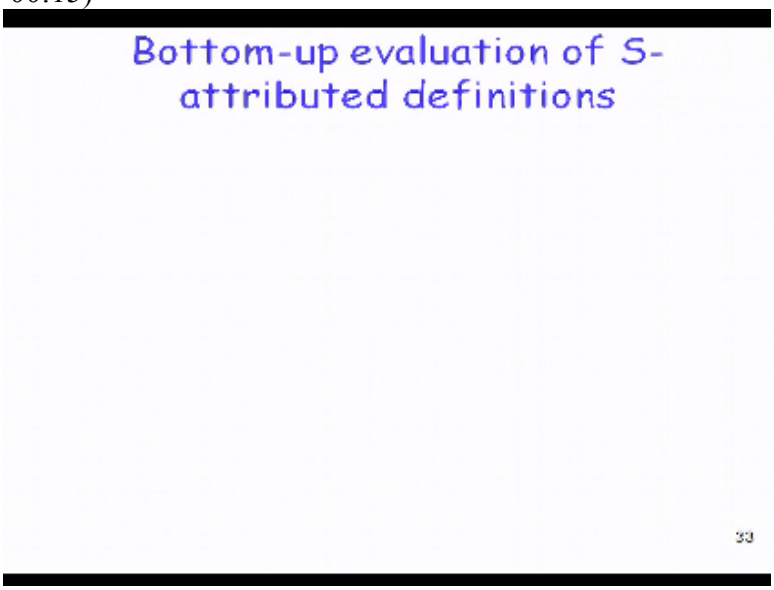
Indian Institute of Technology

Kanpur
NPTEL

National Programme
On
Technology Enhanced Learning
Course Title
Compiler Design
Lecture – 15

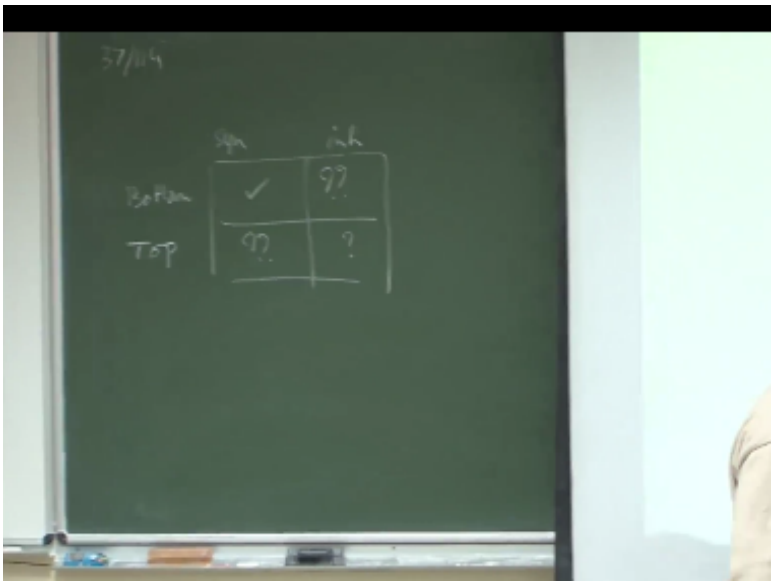
By
Prof. S. K. Aggarwal
Dept. of Computer Science and Engineering.

(Refer Slide Time: 00:15)



So let us show let us part covered discussion for a dealer and the part looking at the matrix were being associate the passing with the concept. The value of that and we look at four possibilities input take that should draw between,

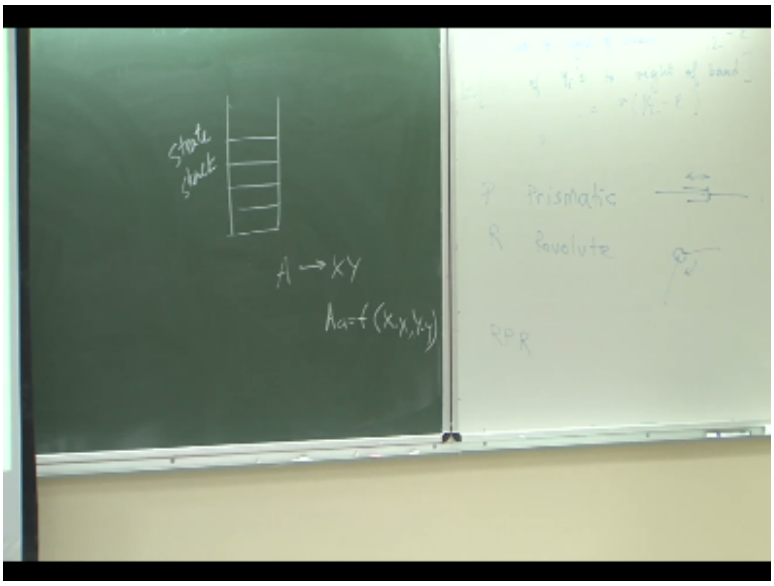
(Refer Slide Time: 00:36)



Can you stop talking now? So we have criticize that reduce and we had direct that the rules and then we look at two part in the assets which of top and bottom of the paladin and at least on one we agreed that when it comes to bottom of our tunics this was something which was simple and when it came to talk down then inherited then we said that at least in some cases we may be able to do it that means I only have to do that so this was slight question mark but top-down and synthesized.

So this was a big question mark and bottom-up and inheritance was since another issue for this access. But since we have two parking at the simple we cannot put restriction on what kind of actuate equations compiler writer man decide to use okay we have to act all this problems ultimately board is that I should able to put act in every box right so let me look at this at least the one which is simplest which is the bottom of and is the size of that the goods only and how device now change my parser. So that I can evaluates houses inside the parsers. Because now we are talking of the implementation. So long with the specifications.

(Refer Slide Time: 02:21)



Now if you go back and recover has something called of state stack? What is state stack contain and bottom of parses states whatever being shifted on this state struck. So we have some state symbol and we get. They have some state symbols are only capturing information their of a implementation where you will not have to worry about gamma symbols at all and we just possible the state symbols because that is the only information you require assume that or logical reasons we have all the information about the grammar symbols of the stacks.

Okay so it is equalent so the elements. Now suppose I have a rule like this it says E goes to X Y this is my grammar rule and corresponding to this I have net to put equations of the form says that contribute A the function of contribute of X and contribute of Y okay. And we were all now simplest so we are talking about SAB definitions. Now we do the equal evolution has you going to reductions now obviously before the reduction happens what was the symbols.

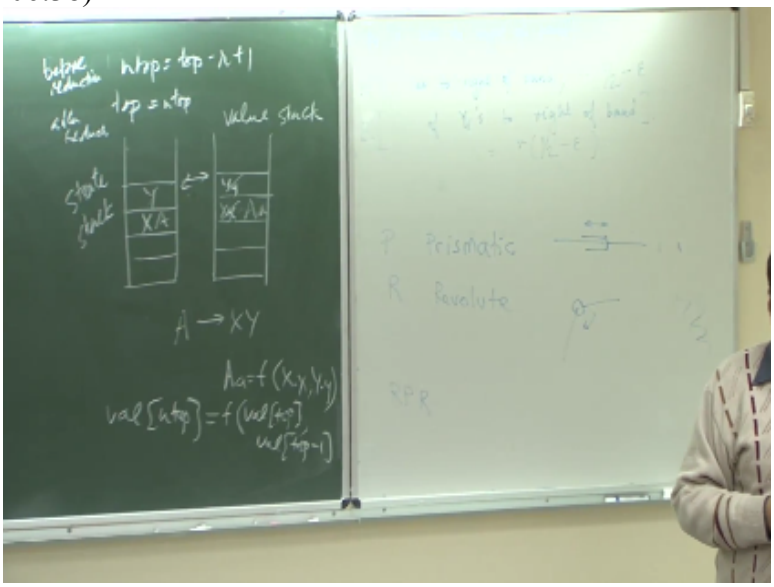
What will be the symbol suppose this is my top of stacks okay bottom what wills the top two entries in the struck before reduction will happen common guys my lord in your figure tips we will right the exam on write this. What will be the symbols in top? Of the state stack before reduction the cause of that for reduction X and Y right and after reductions I am going to swap these two symbols and I am going to push A understand now suppose in addition who states that I also have what we know as a value step.

I am not looking at the implementation in corresponding to every symbols but I can do is I can have four values and the struck the same third pointer that means contribute of X can be capture and contribute of Y can be capture. Before reduction right okay and now we doing the reduction what was I shall do I need to take these two values and then I compute A using this function and then after reduction I just push contribute will stack which is the value stack.

Then my problem is solve because all of equal is contain additional stack make and additional stack which contain all the values of the attributes and the same stack pointer will work for this and as I am pushing the symbol sample seriously in this stack I push the corresponding attribute. So how is my equation look that now I can say that if this is the kind of equation I have this is the kind of active equation I have for this grammar rule if I want to write it in terms of values stakes that.

What will I do I will say that suppose I am computing now value which is when you connect the new top which will come here. How can we compute that I am just going to say that this is nothing but a function of value at the top and value at top -1? And what is new top the new top obviously is,

(Refer Slide Time: 06:38)

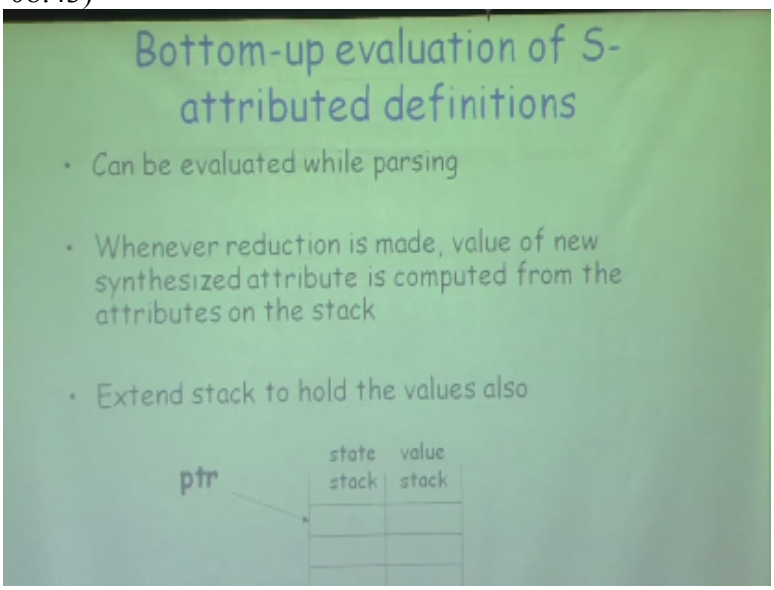


Before reductions if I take top and if I then say that then I have to decide to R then I have reduce this tops by R and then I to advance to this so this is my stack come to the walls and this less this two then I subtract two from this stack pointer and wall to this push this value of A here and after reduction what happens top becomes so this is before reduction. And this is after reduction of becomes in equal now.

So now I can take all my attribute equations and I can give an implementation in terms of value stack of. And now suppose they are symbols which do not have a corresponding active what will I do just sudden value right. So this is to begin with this is empty if I find there is no corresponding at the two digest we can empty. So we that I have same stakes in manipulating State time evolution yes or no this is clear have implement at least this part like for synthesized.

Distributes and bottom up parser how can I implement this okay now some parts of it were provide building facilities for this okay. So like parser yeah like yet will provide you all these $ mechanism but basically this is implementation okay. So this is how it happens that I am going to evaluate.

(Refer Slide Time: 08:43)

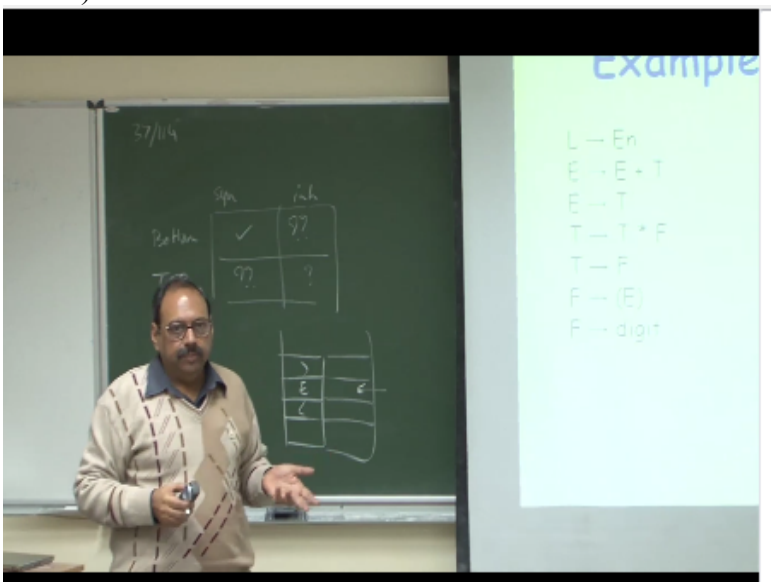


All these attributes while and parsing and whenever reduction is waived value of the new synthesized that is computed from the attributes of the right hand side which are already understood okay. And then all we are going to do is expend the stack to hold the values so I have a state stack and how the values that states that we have already seen were we doing a part of parsing and when you stack is what we use now and top of stack okay.

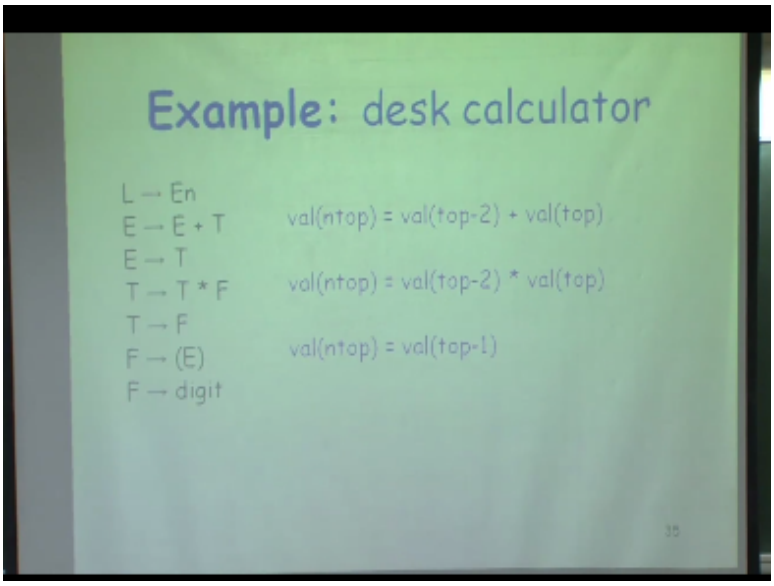
Is and then we say that if the kind of attribute equation I have and this is associated with some production then before the using X Y Z with the value of Z is in the top okay and value of I top – X is top – 2 right this is how it is going to right this is to be pushed left hand side lower and the and if symbol has lower energies undefined and after reduction top is decremented by 2 and state which covers a which is now the left side of the production of the reduction that is pushing.

So if this is the kind neither of grammar I have okay this is the grammar started with suppose I was write all the actions nor in terms of the values okay. How I will do that want to have packages So use the similar project and buy two holes here so that you have already seen this example of this calculator that is when I do reduction by this that means value of expression when I do this reduction what will here at the top of the stack of that will fit in and everything to the start symbol and therefore the value stack corresponding to that must have a U of M okay.

So what kind of action cool side right here and what about this when I say there is only aspect then why'd you do it moves heavy chain in terms of the so he comes off as if a value stands in the stack pointer and then keep adding these values on the step okay so what about this okay let us see this and I have this rule? values pack does not change course symbols let we symbols on the right hand side top value is so new top this top - 1 so this is my stack right,
(Refer Slide Time: 11:56)



This is where the value of ease right and after I pop this and this value should appear here okay so value of end of is going to be value of top - 1 that is before reduction and after reduction top is same as in top right and will be end top, end top will be - 3 + 1 that is top - 2 here so that is where the value will come after reduction the new value will be here because of the reduction all these three symbols are going to be top and F is going to be pushed right.
(Refer Slide Time: 12:43)



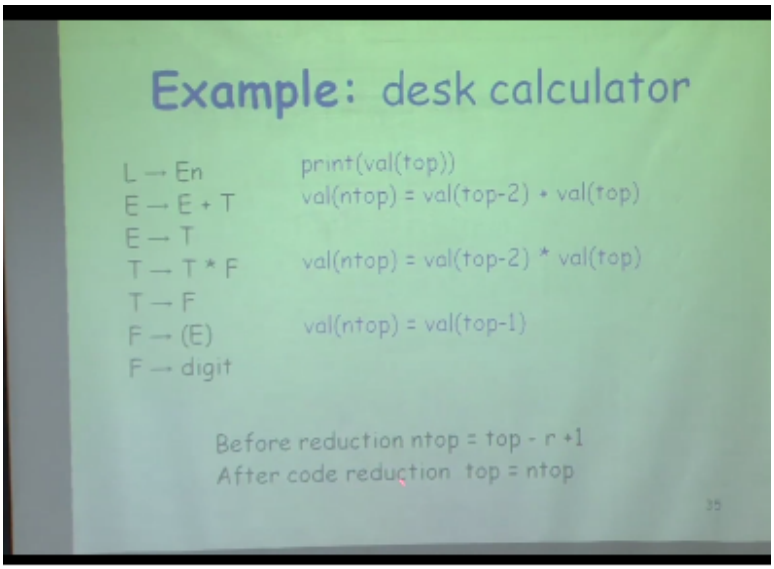
So this is how goose will appear that if I say for bracketed expression value of N top is going to be value of top – 1which is coming the following this because right bracket it is a top this is the top - 1 this is a top – 2 okay. What about this here I say that is nothing but I will have to take top and I like to take top, and I would take top - 2 and multiply to ends okay and what about this same thing but the multiplication will be replaced by tradition.

And what about desk end of this is so what about deskern of this equal so had to anything here so I did not write any reduction so what is happening here that my stack is containing T and the value of T I replace this 5 okay the value is the same so I do not write reduction it off so remember is be fault the act solved $ is never resign anything there is a single reduction my default by the fault what is the value of solve the $ bound right.

That is before that you know that right connection for that okay so for the rest of a I would try any value and here I am saying that whatever is cut off of this act. So this is a unit reduction this will unit reduction and all that is happening is that whenever is pushing being on the stack okay have pushing be it well that is the only act we do on the value stack. And fact this is become self what we were do nothing because now it is replaced by a and the only thing nothing is that all the stack point is do same thing were the value of being this rule.

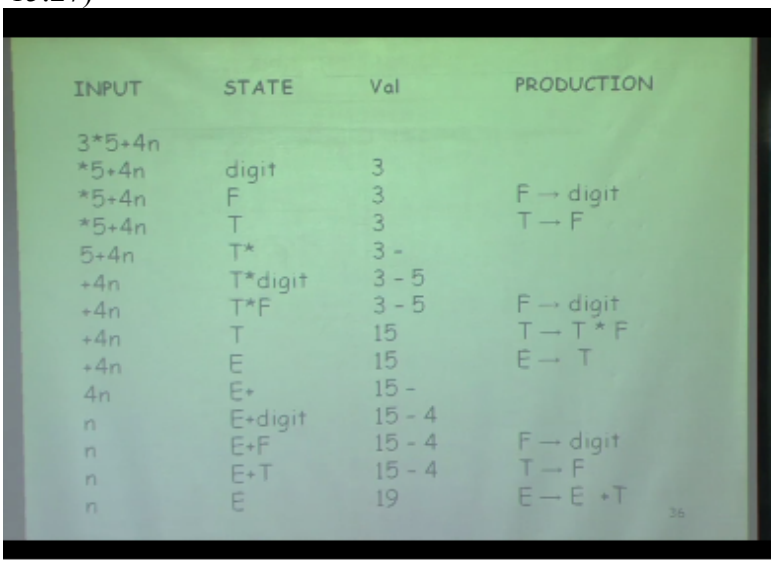
So you can assume that being with my stack complete goodly the values. So I start I do not change that value, so all everything is fine that is it. Whenever I will find the while to be usual otherwise began some value. So I will never rushes right so implementation clear okay.

(Refer Slide Time: 15:19)



So before reduction top is N top is top – r+1 after code reduction top = n top okay and here is an example.

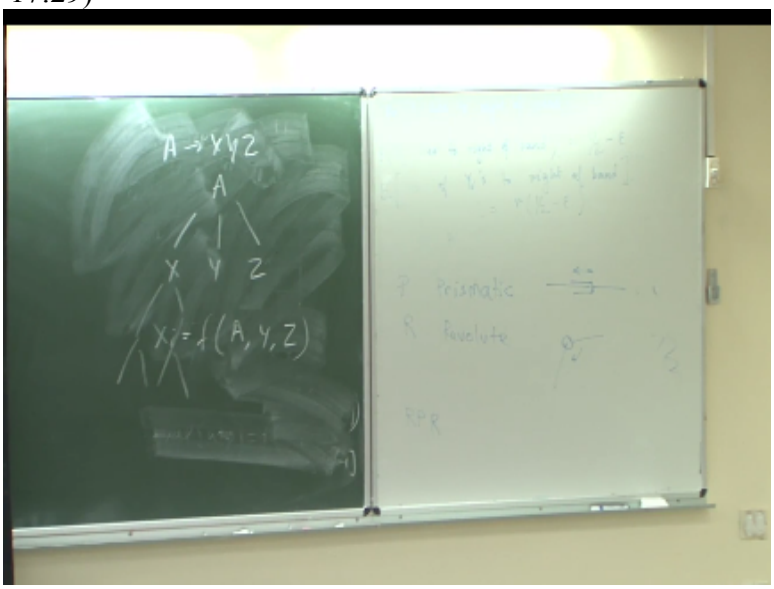
(Refer Slide Time: 15:27)



We can go through this example of saying if this is the we have and if I want to look at by state strut and value stack and the product that is you use it since you have already seen you can see quickly that first I am going to fully digital this stake stack and the value is going to be three so this is the input which is going to be pushed here and this is the register and then I say did it goes to where and 3d means here and this is the reduction and you can see that I have not used any good here okay.

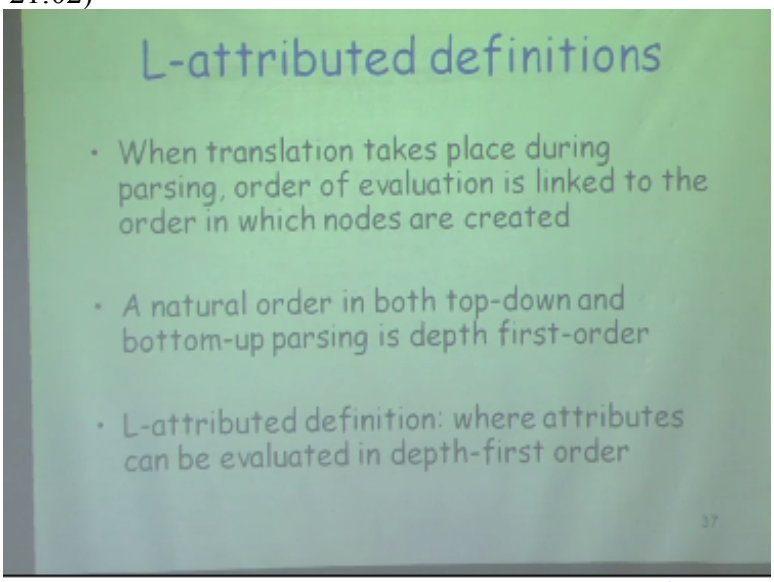
And then I can say that I have equals well so he comes on the stack and he remains there and then star gets pushed on the stack and corresponding to that value stagnant is a value is

undefined so that is undefined on - and then again digit comes from the stack so 3 - 5 is the value stack and then when I do a reduction by this rule when this it was when my value stack remains the same and then I will say that T * F reduced so what will I do now you say that 3 - 5 are going to be pop and 15 is going to be pushed on the side.
So that is becomes value of T okay and in the same manner and finally I am going to get value of E of the stack which is going to be 19 and when final reduction happen says elbows to these new line simple is going to just decide exactly what about that thing this clear to everyone this part is clear. now let's come to this box where we say that I have a top-down parser and inherited items and we started some description on this and let me then just try to refresh your memories on what we are trying to here.
(Refer Slide Time: 17:29)



So first thing it has appear here for the use of top down to parser and dealing with goods okay Then I should able to parser and I should able to sintering value accept a small scorch and the scorch is when I say that A is having production of this form A goes to X Y Z top-down and in pop down parcel this is how might pass these windows and now if I look at inherited attributes the way I am defining my energy that you loops are that inherited attribute root of a simple can be defined in terms of inherited that you to profits.
That means potentially I can now write an equation which says X arritate is a function of A Y and then some attributes of A Y and Z some arrtributes of A Y Z okay. That is have my arrtribute definition consent so this is the valued equations accept the cash that even when I start constructing my positive what will happen is that I said that I am going to expand the leftmost non-terminal all the time.

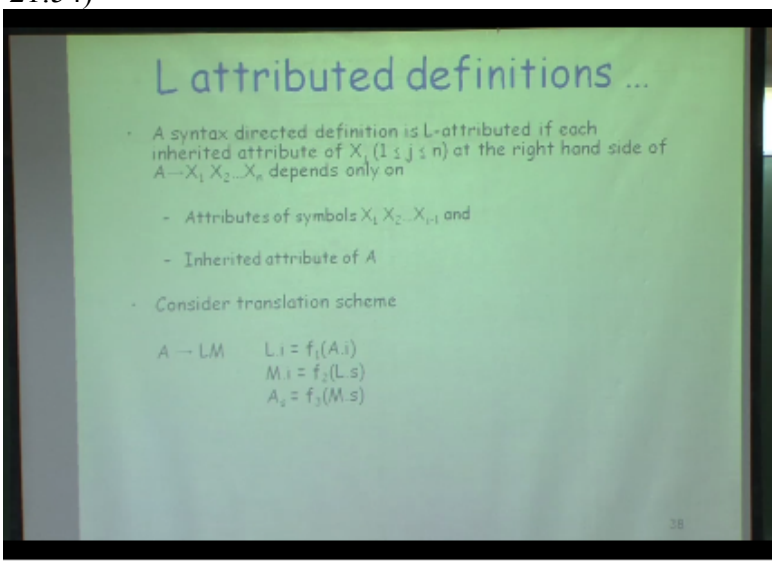
Therefore before I go and start expanding this part I will say let me go and expand this a and I will keep expanding this tree till I eat the leaf nodes and then only I will summon start expressions now if I say that I want to compute inherently time to Duke of X in terms of some extra book of one a node corresponding Y has not even now suppose I restrict my definition Such that if I have an equation like this or the mean slightly change it.
Suppose my I have an acute equation of this form and now I say that for any X I change if I am computing attributes it can only be a function of inherited attribute of A and any attribute of any symbol on the left hand side that means X 1 up to X i- 1 now also what is important to note here is I am saying that as far as parent node is concerned I can only use inherited attribute of the parent node and as far as X 1 to X I minus 1 are concerned I induce either inherited or synthesized attributes.
Then does not matter in top sound parcel if I restrict my equations to this form then I will always be able to compute X I. I never get into a problem because I am now saying that all my attributes have already been computed remember that I cannot violate my dependence. So these are known as electrical definitions so now I am restricting my class of inherited attributes to tell you the Google definitions.
(Refer Slide Time: 21:02)



And say that I can compute this in certain and then and when vacillation takes place moving parts inaudible evaluation is going to be linked to order in which the nodes are created that is mainly the order that I am creating my notes starting from the left and that is really the parsing order and which is the for both whether it is popped on the bottom of that with the first reversal or and if I make this restriction.

Then I say that electrical definitions on whether at once again and saying that if I have only held at you put your definitions then I can do this if I have already would be patients than I can okay so.

(Refer Slide Time: 21:54)

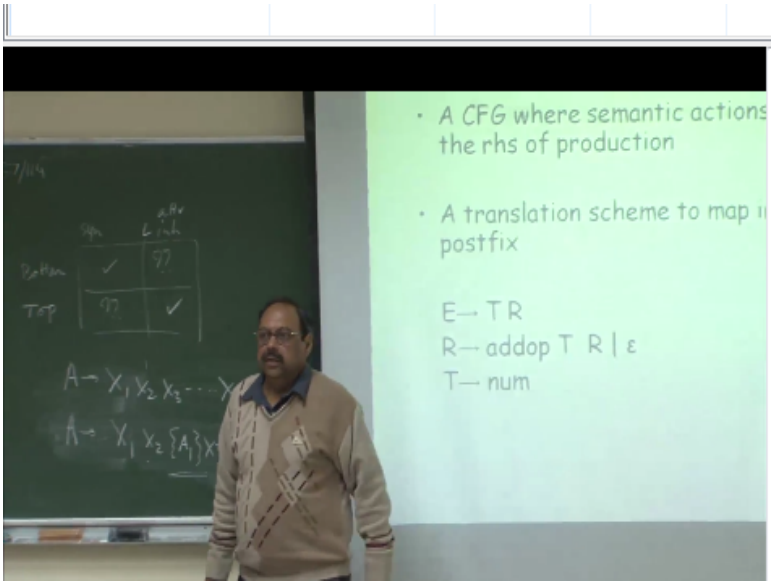


L Attributed definitions are that if I have an arrtribute of production of this form a let me say that attributes of any symbol they depend only on if I take XJ than XJ which is lining somewhere between 1 and end the inherited attribute of XJ depends only on inherited attribute of A and attributes of X 1 to X G – 1 this is K- 1 font is not clear. So now if we look at the past okay so if you look at let us say this so here is a production and here are some attribute equations so first question is this.

So let us look at this if I say le inherited is a function of a inherited that is fine does not violate anything then I say M inherited is a function of L synthesized that okay yes no that is fine because it is looking at the root of left hand side does not matter where the left signal this is in terms of the so please remember definition once again I am saying that inherited attribute of the parity and all that he poops off the left siblings okay whether it is inherit is or synthesis does not matter.

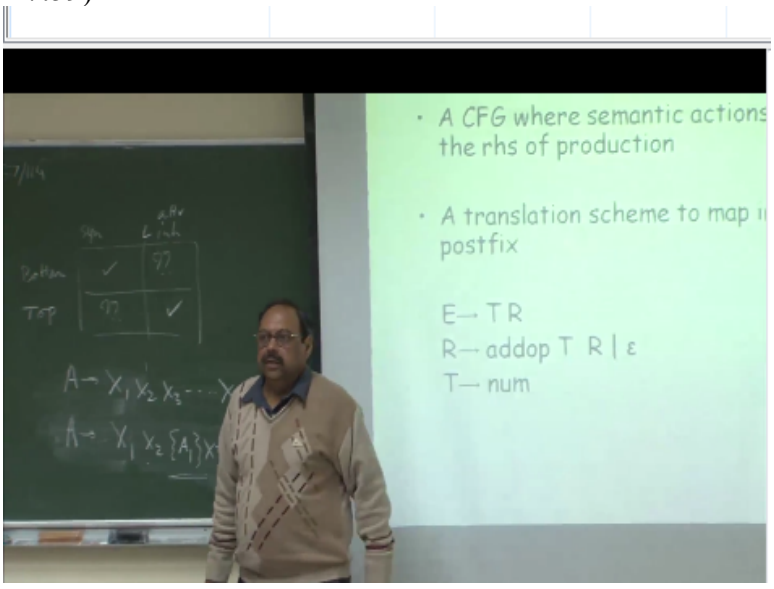
So this is fine if I look at a synthesized which is a function of M synthesized that also is okay no problem so this definition is what about this if I have a rule like this which says if a production says it goes to Q R and there are an attribute equation which says are inherited is a function of A arritate that okay any violations now what about this Q inheritance is a function of our synthesized nobody's fault it is a violation of electrical bill definition because now inherited attribute of Q is being defined in terms of attributes right side okay.

It so this is not a little to technology and if I look at a synthesized which is a function of Q synthesized that is fine now you can see that as far as in fiber strict my inherited attributes to and attributes if then I can even do this then I can write a top-down parser by restricting this so although this was a question mark but we know that now this can be at least on that filling up this problem is born that I am dealing with the situation where I have certain equations one was I had certain attitudes which have only specifications.
One I only had attribute equations but I also said that we are going to have something known as completion scheme where an attribute equations but I will try to impose an evaluation order on those as we do the quizzes and these are going to be Pro specifications but then we can try to capture implementation something so this thing upon translation schemes and basically translation scheme is nothing but same attribute equations but now I am putting these that we do the equations not just from the right hand side like some specifications.
But I will put them anywhere on the right hand side between the grammar symbols in so I can have something like this that I can find a production like this and then I can say a translation scheme may have X1 X 2 and then some action 1 and X 3 and so on so you can see that again I am forcing some action which is a pole or which is an attribute equation in the middle of the grammar symbols not just the right inside but let us see what power what traditional power.
It gives me because now I am saying that I am actually doing something with the implementation now way I am doing this or so what do you a shit for doing this is I say that as I go through this creation of the power stream then at this point of time I do not want to wait till I have seen all the symbols to evaluate electrocute equation I want to evaluate this attribute equation right here as soon as I inform them so really forcing anywhere is normal they say that certain points of time.
So here is a small grammar and can you see the strings which have been generated by this llama you can see that my spot symbol is,
(Refer Slide Time: 27:18)



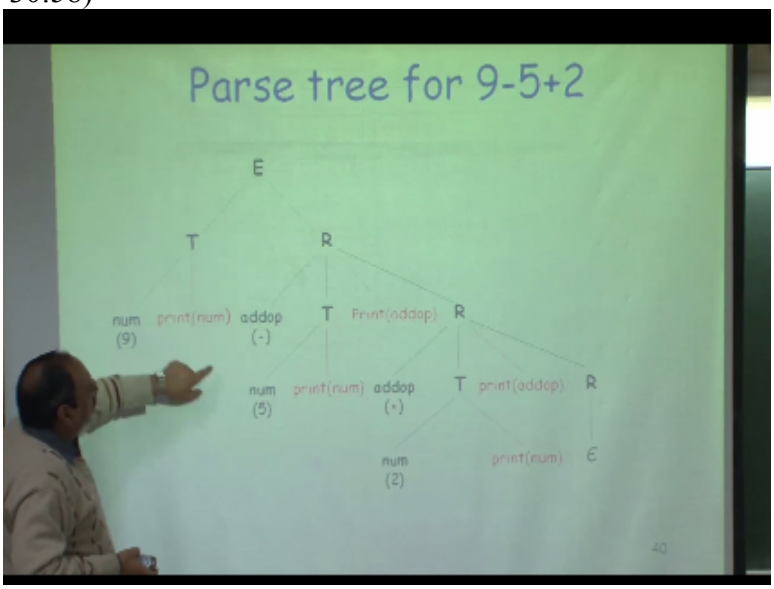
A term followed by rest of it and term is a number and what is the rest of it rest of it is some operator which is an additional operation followed by a thumb and followed by rest of it okay so you can see that this is a right because the grammar is the right III and finally how did I terminate my production by saying that our ghost will sign ok so what it gives me up if I say add up is plus or minus this is giving me some in fixed expression this is creating now a parts before expression. Where numbers are separated by plus or minus so really you can write things like.

(Refer Slide Time: 27:59)



This coat - fighters in there I am assuming air competent it was not these are the kind of strings which can be generated by this and it is s easy to see that what will be the past before this will be equal to P and R and P will go boom then 3r will be that + T and this T will go to 4 and then this

will be – T and R and this T will equal to 5 and this R will go to left side. c12 and so one kind of problem we looked at was that when we reach at the top and we reduce by this symbol at that point is fine you want to say that.
Whatever is the value that should be printed long I am changing the problem slightly and I am changing the problem to say that whenever I reach the top whenever I do this reduction at that point to find equivalent in face instead of infants that is it postfix expression so what are the positives of this do is take this expression tree and then just do post order traversal so this is the postfix and what I want to know is that I want to now whenever I reach the start symbol I want to print this not the bend okay so there is another problem for solving certain problems.
So what we do here now is that we want to use the mechanism of translation scheme and I want to add certain actions in the middle of the ground so let us create some space for adding action and let me add an action here which I say that I want to think adoption and then I'm adding another action now actually evolution on between two gamma T it's so I've added two actions here one is here which says bring that up and another reason exists right now yes now that we take you to implementation of this.
So what you want to do to begin with is that let me say that each of these actions is going to be treated either the grammar symbol of but it is treated like a growl symbol and we create my pasty in the same manner it so if I now say that I am on two parts let us say 9-5+2 and I want to print postage for this first I will thing I do,
(Refer Slide Time: 30:58)



Is I will create this parcels okay so this is the past before 9-5+2 now if I say that this is also being treated as a leaf node token then my pasta will change and my positive is changed by putting this

action because now I am saying that T goes numb followed by this print num print now is at a leaf node so this is how it will change that T goes to numb and print now at this point of time I am not executing any action but I am saying that this is the and similarly I will put for all these a print num and for all these productions.

Which says R goes to - PR I had introduced in action here which said R goes to - P print at all followed by up and up here and up in that this is how much positivity okay so I have taken this party and I have introduced all spawning actions also as equals yes now suppose I lose our depth-first traversal of this tree and whenever I encounter responding to an action I thought to happen off what only the first node that will get printed 9 then 5 then - then – then 2 and then + the post-its Sakamoto speaks of the river so not using slightly different.

Mechanism and I keep talking about implementation I am saying that because translation scheme because at this part of by no means we really do not want to create this parsley and then have another traversal whole idea was that I should be able to do all these translations as I am parsing so i do not want to create parse tree first and then say that these dummy nodes are actually part of the parse tree then traverse parse tree once again and then excuse these dominoes. That is really not that that is only the logic is really to go into the implementation now have this implementation in.

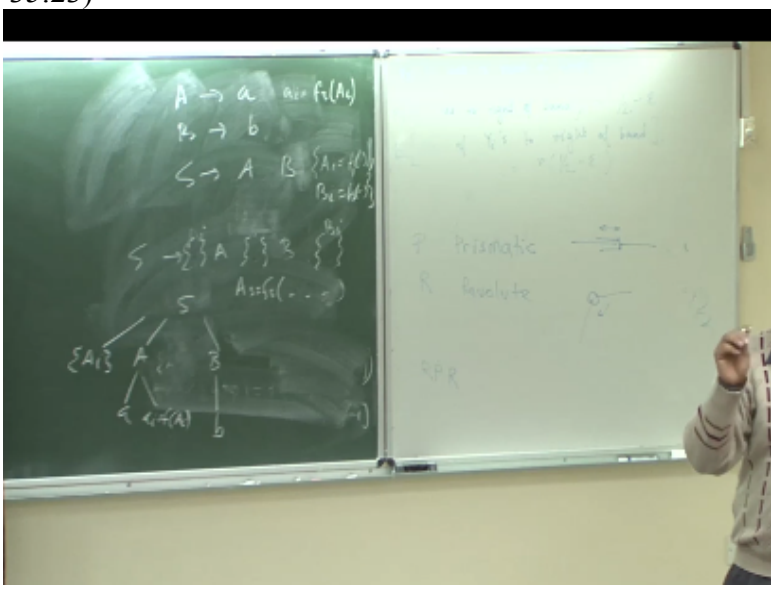
So what we are assuming here of actions are nothing but terminal symbols and when i perform a depth-first traversal order to obtain when I performed at festivals an order whenever I reach the section that whenever we reach this terminal symbol corresponding action is going to get executed and when we design translation schemes. We therefore have to ensure that when I say that print num here this value of nom is available okay suppose.

I put it at around place and value of num is not available then what will happen my data difference is about we get value dependencies remember so that we had violated if I do not put my action directly it so when I said for example I was trying to parse an expression grammar of trying to pass Nor bit what was that round the thought and expression girl that was a declaration write a declaration of I followed by a list okay.

So when I was trying to pass that stream in some way I said and inherited is p-type now Ellen attitude this p-type rather dependence there should say that it depends on P type and it has been evaluated I really cannot evaluate this equation so this equation cannot be arbitrary evaluated at any point of time but it can be evaluated only at the position becomes available Hey so for example if I say print num here instead of this print num I put it somewhere else by them then you have may have changed so that will really be not the right place.

So finding out the right place is very important in translation sees now let me pick up another example,

(Refer Slide Time: 35:23)



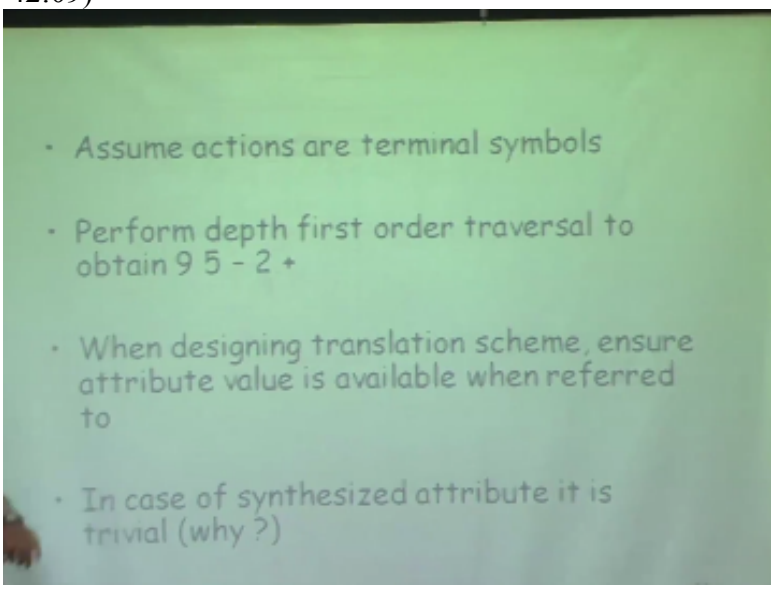
So suppose I have a production of this form and I want to compute let us say I have two equations with this and some computation of inherited which is function of something you inherited because now I want to put an action here I have multiple places where I get protections so if I say s goes to a B where should I have these feed places. Where I get protections where should I put inherited computation this foxes this box of this box and you also have to give me a body.

Why we are putting it middle one this one this one right okay so this is where I can be a variable and where do I compute be edited yes right this is okay everyone agrees with this it is very good so now let us start to make a parse tree and let us see whether it works or not so suppose D and B are also non terminals in general they can be so let us say that do not have productions which says it was A – a, B – b, and then I say that here so I am not compute some active I can compute attribute for the same okay.

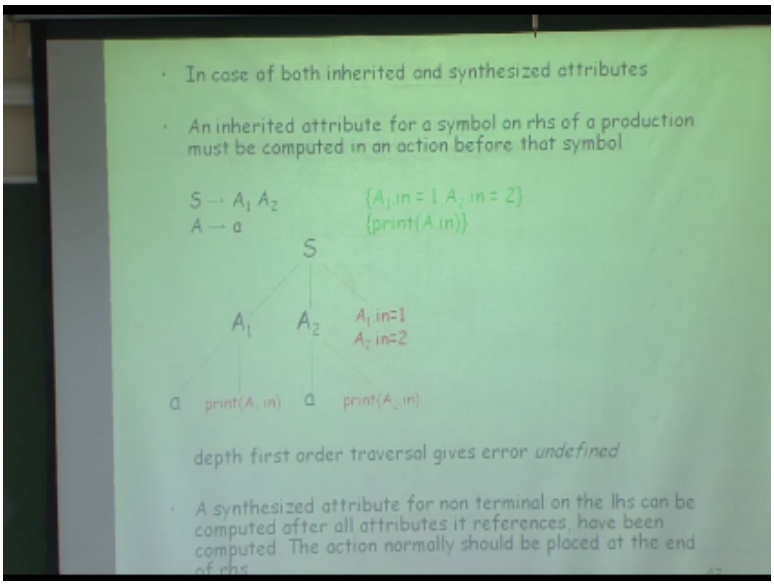
So now I compute an attribute for this a and I say that this is a function of a very is that equation so if that is correct that what will happen let us look at my party my first we will look something like this I have game I have B this goes to Lane this goes to B these are the kind of swings are generated and then I say here and computing inherited right this is between each and now I say an equation corresponding to this is saying that A – a i which is a function of a narrate.

In that posture was a lot of I will reach this load and say any value of a I e but he has not been computed is it this one you see the problem why I am had is the second position now what does

the air depend upon it depends only on the sibling suffix so why I cannot come to inherited here does it prevent me from doing any computation here can I come to paying very pit in yes so if I compute a heritage here okay then I do not have to worry about this problem.
Because in that case what will happen is that I now compute and I said now give me this value okay this value has already been computed so when I am trying to compute a hairy hidden be inherited they must be computed in the place which is immediately to the left of and immediately to the left of because I can synthesize so here or hear what are your faces right off right off the symbol No so suppose I want to compute a synthesized version. After a filly these are after all so if I guess do these computations then I can handle mine at least equations without violating any of the dependence problems okay.
(Refer Slide Time: 42:09)



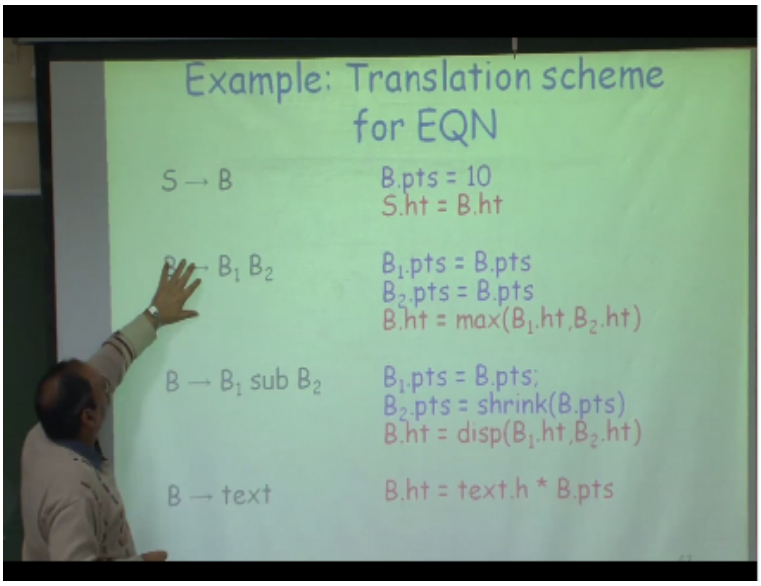
So in case of synthesize that you do this is but in case of both inherited and synthesized attributes inherited attribute of a symbol on the right hand position.
(Refer Slide Time: 42:14)



In case of both in arritation in goods inherited root of symbol in left side must be computed in the action before that signal immediately left of this so if this is the kind of empty boot equations I have you can see that when I make a pass to you like this and I put all attributes in this position when I am saying that I want to print a one inherited a one inherited has not even been initialized and therefore a one inherited.

I have been computed in this position which is immediately on the Left pocket that I will not have get into any of these violations similarly a two inherited must have been computed on the left of it then this equation would have been fine if so depth-first traversal order in this case is going to give an undefined error and synthesized attribute for non terminal on the left hand side can be concluded after all the attributes of the reference have been computed in action. Normally should be placed at the end okay.

(Refer Slide Time: 43:13)



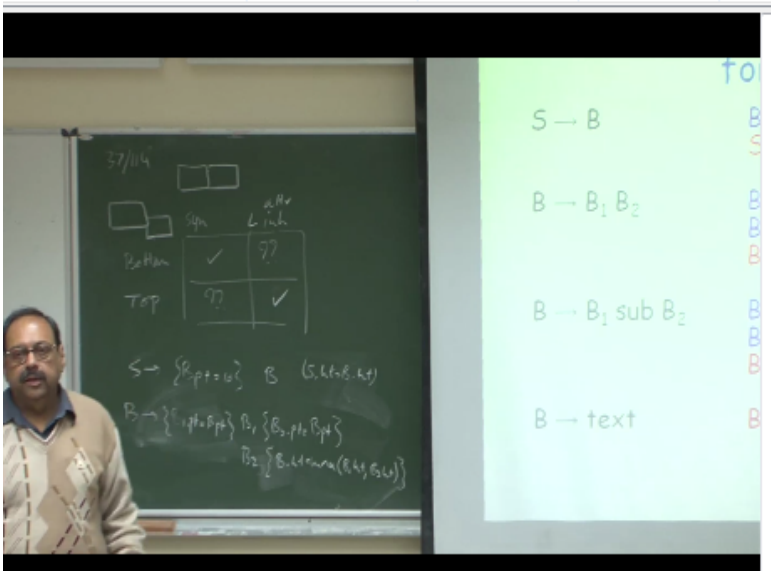
So now let us look at our translation scheme and try to create the transmission scheme and suppose this is the okay let us ask you and ask you to compute the translation scheme alright the translation scheme for this actually is a grammar for equation system which say that start symbol is a block a block can consist of two consecutive blocks or a block can consist of a block and subscript another block really like saying that if I am right trying to move typesetting.

I can have two blocks and I can put them next to each other I can hire a block and I can have a subscript block to this only thing that happens in case of sub script block is then the font size of this will be smaller so you can say that subscript is going to be 75 percent smaller than this font size and this will be slightly replaced or displaced okay so that now the total height of the block chain is and then the kind of computation I am trying to do that if I say that my quantize is an inherited good.

So I have to kind of attributes now so let me remove since says that if I say that my font size is an inherent attribute so when I start doing typesetting system I say that I want to move types at this document since then font size we have done enough of a bit to know this that right in the bacon to declare the font size and then you say that p1 point size and v2 point size are determined by this. So b1 point size is V point size and three points.

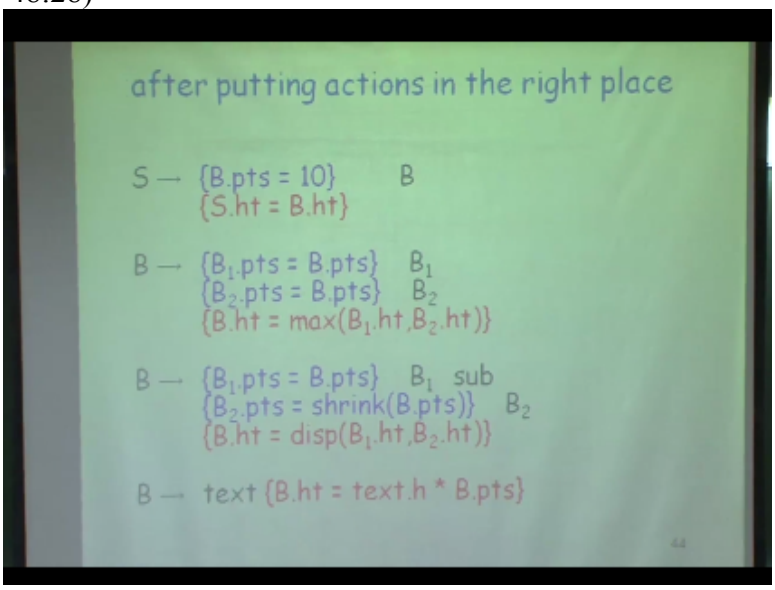
but when it comes to a rule like this which says that B is a block which consists of B 1 and a subscript Block B poop then even point size is same as B point size but people point sizes slightly smaller and depending on your title setting system you'll have a shrink at something like 75% 80% 70% does not matter and therefore we use this function now what do we believe the height.

It so then we say that the height of this text is going to be B height is going to be determined by the text height x font size so some function right and then how we like determine height of this B now you can see that I have height of B 1 and I have height of B goo but slight displacement so total height will actually go up and therefore we say that B height is nothing but with some displacement okay so you one can determine what is the displacement factor.
Whether you want to align it in the middle and so on okay but the total height is going to be more than height of B 1 and B 2 and therefore I say that we hide this displacement function what about D height here B height is going to be simple that whatever is the max of the q u assuming that at least there is going to be a language one who reaches and then what is the height off the spot symbol or the whole text then that is going to be the behavior then.
What this into or translation see that means I want to put all the attribute equations in a place that I will not have any violation of evaluation whenever I am trying to do that computation all that attitude it depends upon will be available no such violations you take this where should I put all these that you put synthesized attribute will remain in the right position that you agree so where should I put this equation we find right in the translation scheme. Where will this book the point sighs is a fine time left of me so my attribute equation should look.
(Refer Slide Time: 47:03)



Something like this after then converting this into a translation scheme it should be something like think that the point site is then that becomes the action and then I have B and then I say that s height is right this is our key what about this all this equation change this school and these equations in terms of translation scheme of Italy loop Vigo school be one point spice is fine be

point size followed by v1 followed by two points is. Fine D point size four or five be 2 followed by this becomes my position right okay similarly so now I can do the same thing here,
(Refer Slide Time: 48:28)



I value it on the left hand side so the way to read this is left to right top to bottom so I usually even on left off even one size in length of v1 we do point sizes on left of t2 and then this at the end and similarly B 1 point size on the left of v1 and we do point size on that would be 2 and this and then this is my transmission now let me all the questions you okay.
Suppose I instead of this I say you got a little kit but I always be able to do this evaluation. A very good sorry right here right in this case it may not matter but we cannot have is delusion. Therefore rather than looking at court with attribute which depends upon we have a gentleman we say that concludes all his lovely Peter Kaufman Ramos single so you do not have to look at what does it depend upon because in this case now you are saying oh it depends only on the inherited execute and therefore of the parent in therefore I could have concluded okay but then will be changed from case.
To case and therefore I do not want to take that kind of analysis and I will say that always compute this equation on left of B 1 and this equation on the left of B 2 and so on now that we understand transmission schemes okay we are going to attack these two problems we are going to attack this box and this box and say that with the help of transmission speeds can I now use electrical field definitions for a bottom up parser and synthesize definitions for a top right these two boxes for left in these two we have understood so now we are time to stop today and this will take off in the week there is follow the measure me okay.

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 Lals
an IIT Kanpur Production