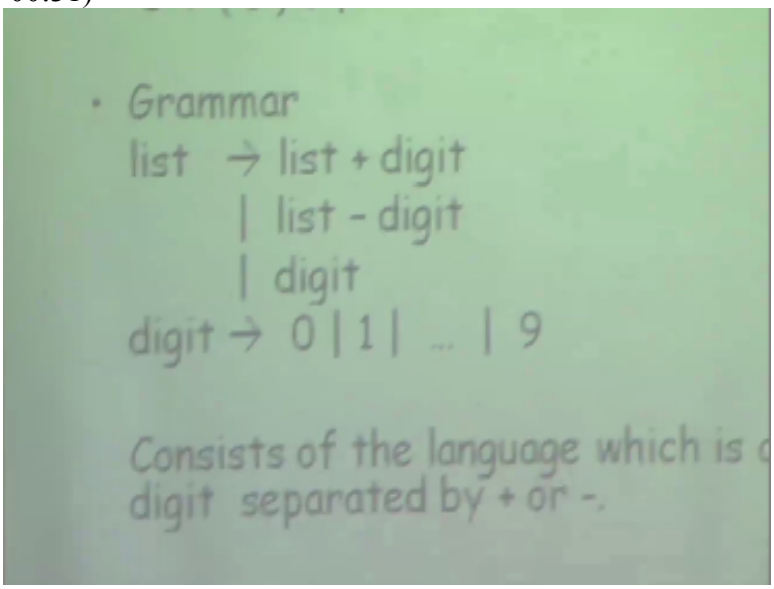
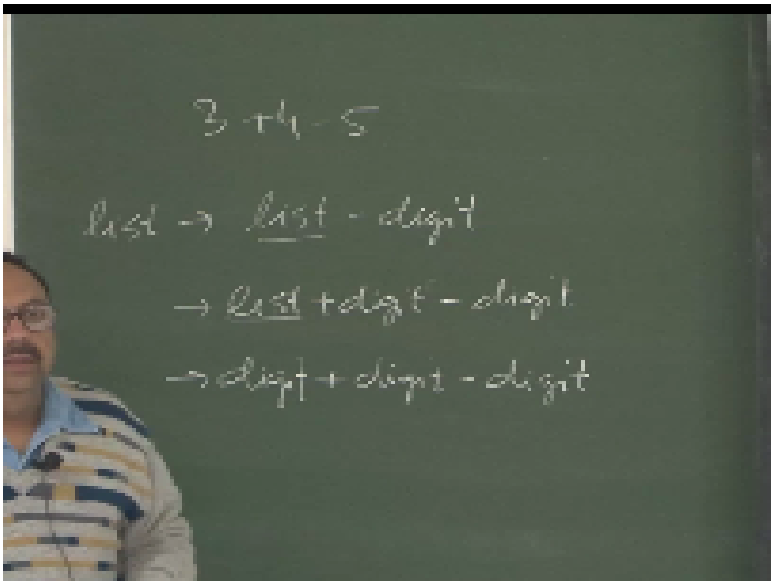
Indian Institute of Technology
Kanpur
NP – TEL
National Programme
On
Technology Enhanced Learning
Course Title
Compiler Design
Lecture – 07
By…
Prof. S. K. Aggarwal.
Dept. of Computer Science and Engineering.

So we start with looking at how you use grammars and how we specified index use grammars and there were two examples, one was how to specify pedal set of parses.
(Refer Slide Time: 00:31)



This was grammar for penal set of pansies and we had another grammar. I am going to look at This is that if this is my start symbol and then next thing was that how you want to check even a string in either is language or language one-way to look at known are right hand side so let us look at examples that suppose I want to check whether .
(Refer Slide Time: 01:15)



This valid speaking language specified if it check it is my startup symbol Can I derive it from the stock stable and derivation means that I am going to say that this is a non-terminal, which gives me one of these rules and if I replace it by one of the rules and continuously keep on doing what I suppose somehow so let us say that suppose I say that this is replaced by this minus list and also that I noticed that this is our terminal this is a terminal but this is a non terminal so I decided to replace this by again what if the rule .I have and this time.

I decide to use the first rule and that gives me this goes two and then the remaining part remains as it is and then I decide that since this is the non terminal at this point of time I want to use another route so I decided that I want to use. The third rule of the production and replace that by and by a digit and now I find that this is the non-terminal and now I want to start using one of the rules here for a placement and then I will go around and say that this first digit replaced by let us say a production which says digit goes to three it goes to three.

So this gives me and then I decide that, I want to replace this particular non-terminal by a game one of its productions and this times the production .I want to use it says digit goes to four , so I will have and now this is the only non terminal that is done and I want to use a production which says digit goes to five and if I do that once I get string, I can say that this particular string belongs to the language.

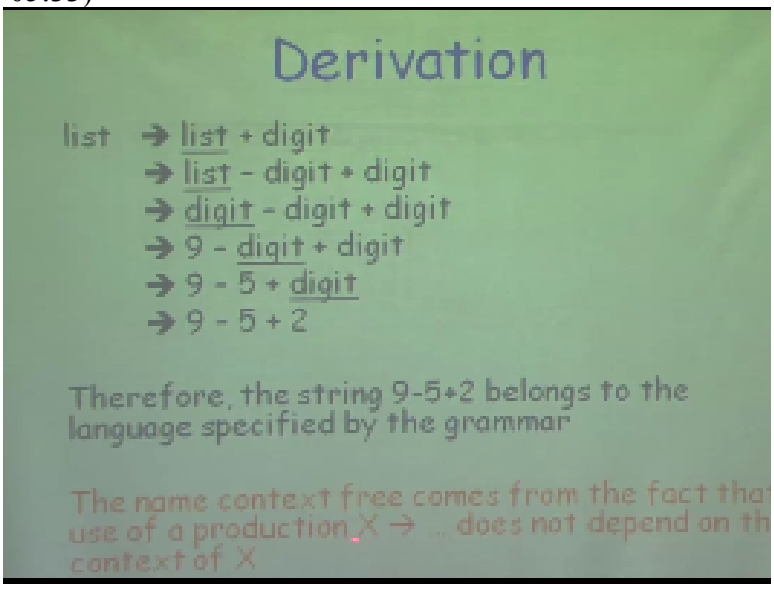
Which is specified by one and the several things left out of unspecified okay, I was saying that so you can see that I had list going to list minus digit why did I choose this rule not some other rule okay, which rule to pick up some how there was a magic company and always able to pick the derived rule second thing that I had an option here of saying that this digit could have been replaced by five and I would have continued rest of the derivation so in which order.

I have the place that was not specified so which rule to pick up that is left unspecified which simple but somehow I managed to pick up the right ones and reach this and final answer is that the string belongs to the right now for process of parsing it is very important that not only we give this answer which is saying yes or no because it is possible that at some point of time. I picked up either the wrong symbol or pick up the production okay, than then it is possible that although this is a valid string.

I would not have reached that conclusion okay, so it is important to know in that big symbol to pick up and which production to pick up just giving this answer that this particular string belongs to the language specified by this because it can be derived from the stocks symbol it not sufficient in the whole process about and picking up the order, and picking up time production.

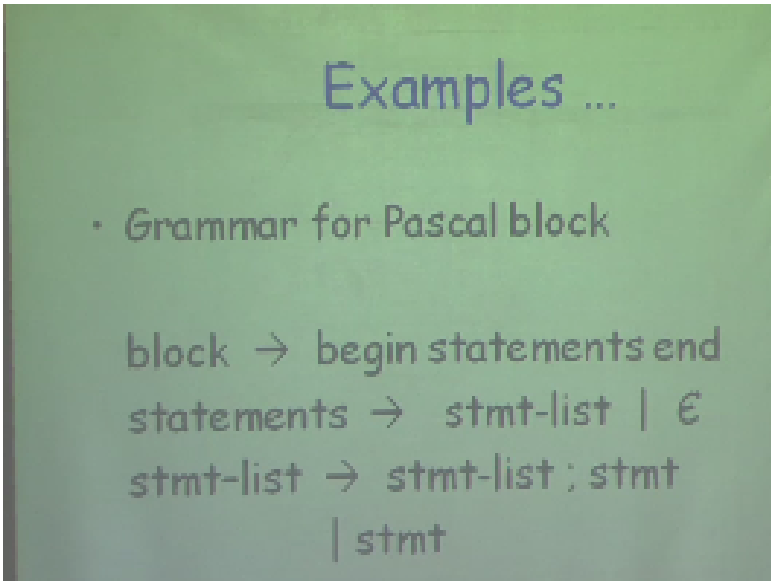
So we need to understand how to do that so let us move on so okay so here is another example I just picked up.

(Refer Slide Time: 05:33)



So I am trying to show that this string 9 - 5 + 2 belongs to the language specified by the grammar and other sequences I have follow in the production and therefore we say that this is this belongs to the language which is specified by this grammar and the word actually losing context-free grammar perhaps you already know it from the fact that I when I look at expansion of any of these non terminals. I did not look at what was in the left and what was in the right .I just look at this symbol

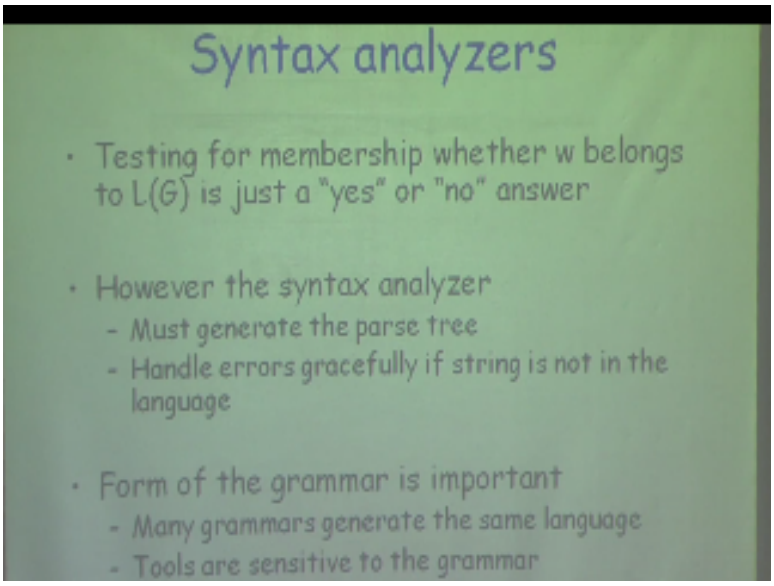
(Refer Slide Time: 06:12)



Which is so there is one more example grammar it is if you are familiar with any block structured language or Pascal this same that a block is something which always has keyword begin and then in the beginning and then you have statements between these two symbols and then these statements actually derive a list of statements or it could be null that means a block would discuss two keywords beginning and end and then the statement list arise now this is a recursive rule it says that statement list derives.

A statement list followed by a single statement or it just is a statement okay, you can see that what I am doing here is basically saying that a block consists of two key words begin and end and what you have in between our statements and each statement must end with a semicolon except the last this is the these are the set of strings which are being derived by this particular problem because we see slowly building of this telling you how to start specifying programming languages three grammars. I am giving a simple example and slowly will start anything. So is this concept clear at least the part of the revision.

(Refer Slide Time: 07:34)



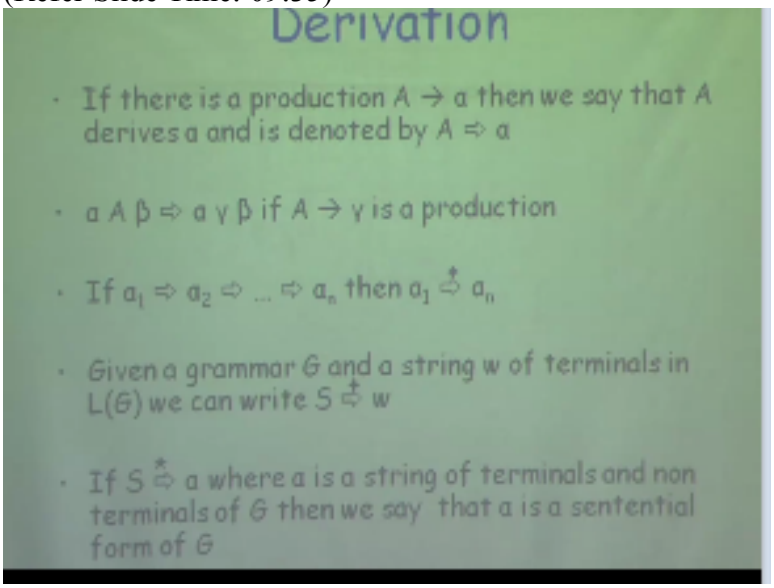
So let us look at few more things and now we say that how do I build syntax analyzer so let me go back and try to give you a similarity with what we build syntax analyzers have you have a set of specifications and then you were talking about certain kind of orderings are in priority for maximal bunch principal and so on so just specific nations are not sufficient we need to do something more to really build up our so testing for membership that whether the string belongs to the language specified even just a yes and no answer this is the base index analyzer, but also the syntax analyzer must do additional things one it must generate a parse tree if I define what parse trees.

But basically pass these caps in which order not doing the derivation and then it must also handle all the errors and when I say gracefully what that means is that it should not just do an order for stop at the first adder but should try to find as many as possible but it should recover from the error and that should continue from that point and important thing here is that the way I write my grammar that is important because I can affect multiple grammars which is specify the same language.

So I can write G 1 and G 2 which have totally different form but when I look at strings which are generated by them they are exactly identical States but the form is important if so whether I pick up G 1 or G 2 for parsing that is important and we see what are the properties ,which will determine whether some grammar is better than other grammar so we look at that and then the tools we talked about so yak is one possible all those tools are going to be sensitive to the way you become so find for example if you are familiar.

How many of you have heard this word angler so you only know Yak, bison so you only know yak.

(Refer Slide Time: 09:35)



Deer to you of the popular name then you can see the So neither species of the same animal on so yak was one of the first pools which has built up and this is also belong to a category of parses important of parses we see are the parses are there is actually of and bison is supposedly so we will see all these in so what is derivation I already showed you a form of derivation so what we say is that if you have a production of this form now look at the form this is equals to α and here rotation.

So when we see use symbols like α what that means is it is not a terminal or continuity normally in this context this is you stand with stream of turbulence and String so this is not just one single but actually this illustrate so this says that when equals to α and we are saying that a device α and the rotation we use here is a take a rope it says that a derives and so normally if you start reading textbooks and if you start reading papers without definition some kind of normal page.

Which has evolved and that is when you have uppercase letters that is normally going to be a non-terminal, when you have the Latin letter α,β,γ more program normally going to be strings of terminals and non-terminals and lowercase letter or something which is like a popcorn on and so on those are going to be comments okay.

And somewhere you will find that there is always region so you have to look at the context and we will try to understand it now we also say that if α A β up derives α γ β okay then a point to γ must be approached because the only way this string is far a β can derive α γ β is that a derives so you can see that it is not directed by the context is on left of it we the right of B and when I replace a /γ then α it is from the electric meter remains to the right so you can see here that when I am saying that this thing is sport.

Whatever is on the left of this type in whatever is on right that is copied as that does not change okay so this is a derivation and this must be production than only this derivation could happen valid and if I say that if I have a signal spread of derivations which says α 1 derives α2 which derives α 3 and so on then okay, so that rotation will complete a so this phase that will forward arrives and fine so I can have a series of derivations and this is actually deriving one of those steps.

So I would like to capture that somewhere so we say that given a grammar G now I am trying to define these things in terms of derivations and we are saying that if I have a grammar G & G has language which is LG then what we can write is that all the strings which are in their LG can be derive in the starts end okay, so this can be done in one or more steps so S derives W if W belongs to L and if S derives α in 0 both S then we say that this string α which is a string of terminals and non-terminals is essential function.

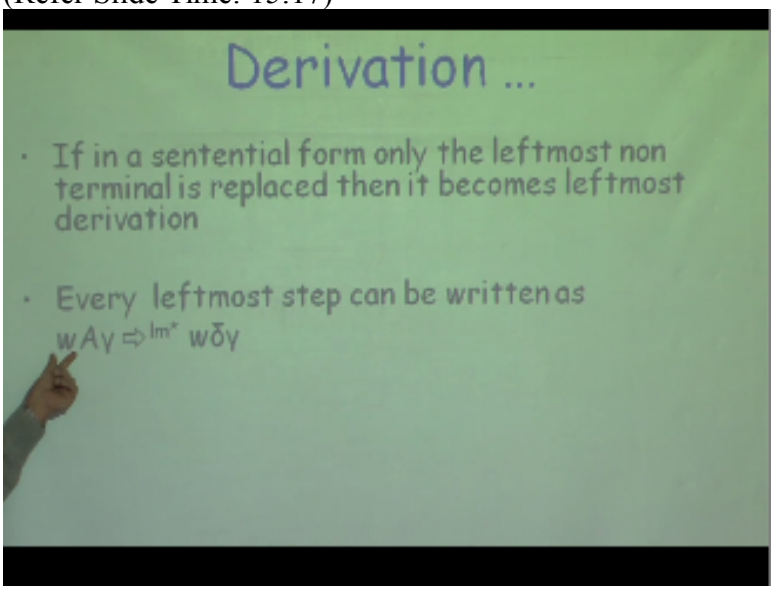
So looking at here if I say list is my stock symbol and list is deriving list - digit okay then we say that this actually is a sentential form of grammar I have okay ,similarly list plus the get minus the digit is also a sentential form of grammar. I have so basically whatever string of terminals and non-terminals can be derived from the start symbol all those are sentential forms of money now this also gives you a him saying that when I start doing parsing and I want to flag errors and whoever it will be at any point of time.

If you reach a situation that something is not a valid sentence at form of the grammar then what will you do you can immediately flag in error and immediately say that this is not a valid says they should fall and so these are some notations you have to be aware of okay, so continuing on this so if in a sentential form okay.

Now let us look at this so let me just catch something I did here ,so one thing I said was I am not worried about to begin with and I did not worried about okay, but actually I was using an order without and what, I was doing was that all the time I was looking at the leftmost non-terminal and I was expanding them so if you see here this is a non-terminal but I chose not to expand this I chose to expand this and when I expected this then again. I decided to expand this and not this and this okay.

So what I was doing was that all the time .I was speaking of the leftmost non-terminal and expanding them till everything got replaced by one of the terminals what I still did not tell you is that how did I pick up do so in derivation if I say that all the time if I have some sequential form and the leftmost non-terminal is replaced then we say that this is the leftmost derivation so what I have shown you is actually.

(Refer Slide Time: 15:17)



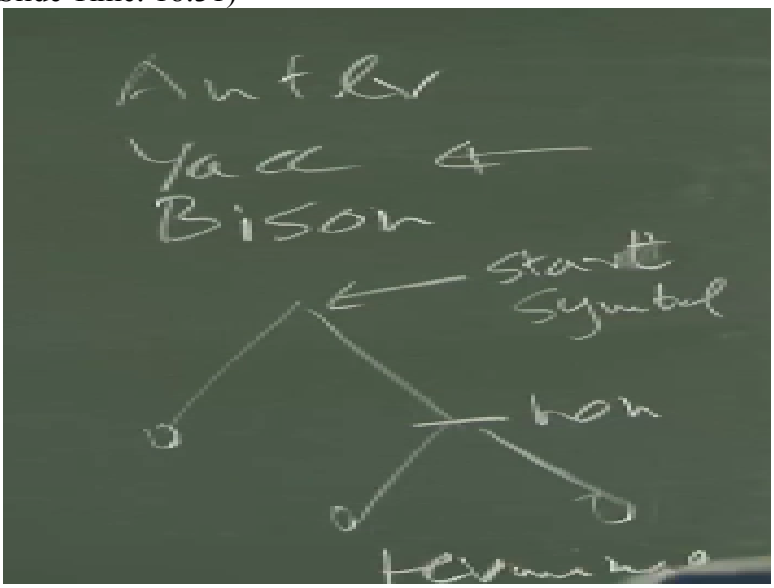
Now similarly I can write each leftmost derivation the notation. I am using here is in derivation. I am saying this is leftmost derivation and what we are saying here is that if I have W a γ and please understand the notation here so W a γ where W is a string of terminals a is a non-terminal and γ is a string off terminals and non-terminals and what we say here is that if I replace a and obviously it going to Δ must be the production's in my language then we say this is left most derivation.

Because a left of A and only have terminal in this right of we have terminals and non-terminals so this actually left most derivation to this notations is that if I keep on doing it left most derivations 0 and more steps okay I keep on generating terminals symbols on lest on side okay, and similarly I can take about a right most derivation like I talked about left most derivation if I always replace my right mot non-terminal one of the symbol of the tokens then this is going to be so they are similar we look at both and hers is now catch okay and here we say that some grammars like.

I said that it is important that you have multiple grammars same language and tools is sensitive tool is sensitive form of a grammar, we say that some grammars are ambiguous because they can produced more than one because a Teri point of time so if you have antiquity we had are problems see what of problems this concept become becoming clear to everyone we are slowly moving towards the parser in this each parser we just try to have two things to design which rule to pick up and which symbol to pick up okay.

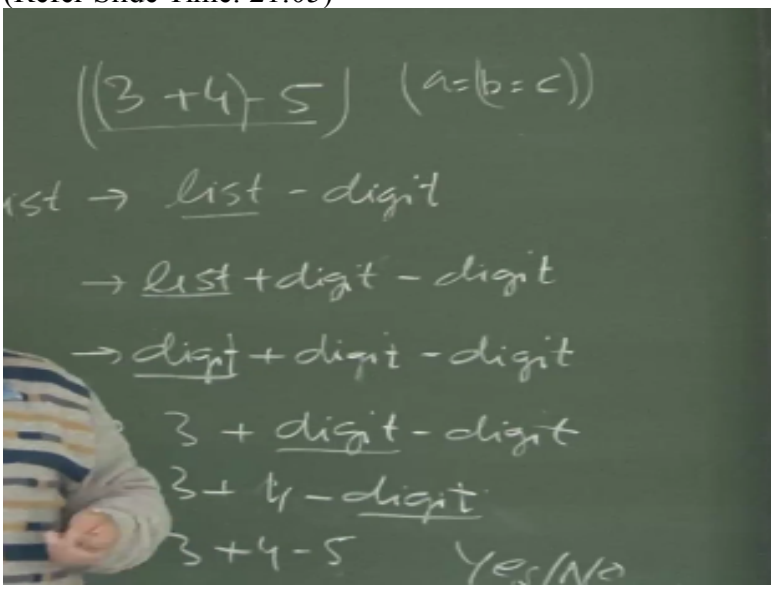
So if I am doing a top down parser and I will design formally parser are than I am looking at left most derivation if I am looking at bottom of parser like I talked about lest most derivation

somehow I am unable to pick up rise looking at certain symbols okay so what I want to do now is I want to slowly move data structure and then after introducing the data structure I will continue with rest of the parser okay so what you want to take now with I want to capture all the information parser so what information I am trying to capture.
I am trying to capture information about what rules are used for this information okay as the name suggest that you have data structure called r 3 then 3 every limited point has specified each of these my job is done and so one of the thoughts energy.
(Refer Slide Time: 18:31)

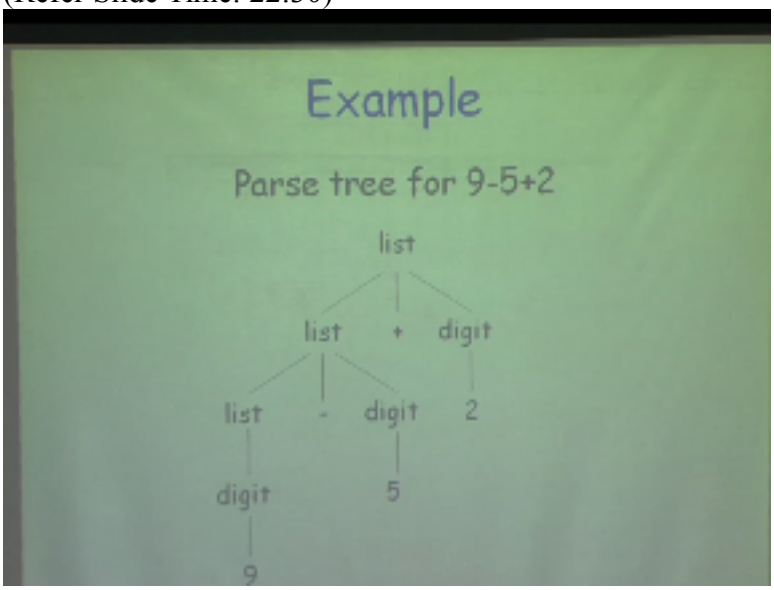


In this case if I say that I am making a tree root node is a node that corresponds to the start soon and leave notes on the notes which was well as nominal sum of dominance and internal nodes in the correspond to set of terminals so this is all I have then I have a stock symbol I have internal nodes which correspond to the non terminals and the leaf nodes which cause onto the terminals I try to look at the derivation here B 4 + 4 - 5 ovulate do that I say that I have this list which corresponds to root note the and then this draw slightly.
So I have this list which device list - so what I do here is I say that corresponding to each of it is right inside symbols and then I have edges which each connect this so I say this actually it is this - video and then I get a derivation of this list and that derivation was saying that this goes to plus B so this gets replaced by list +list and then this list but a derivation so here is a digit and then this is a 3 so this midget goes to 3 and then the next digit possible 4 so this 4 goes to 4 and this digit goes so this is the past before 3 + 4 – 5.
With respective to γ at different γ we need with parser So this shows how start γ symbol derives so staring from here I so example of so this is the in this case and which is this - and I say that

because now if I try to say that I am looking at 3+4-5 showing you all the steps and root is level by the start symbol leaves notes are taken by the tokens and internal notes are non-terminal.
(Refer Slide Time: 21:05)



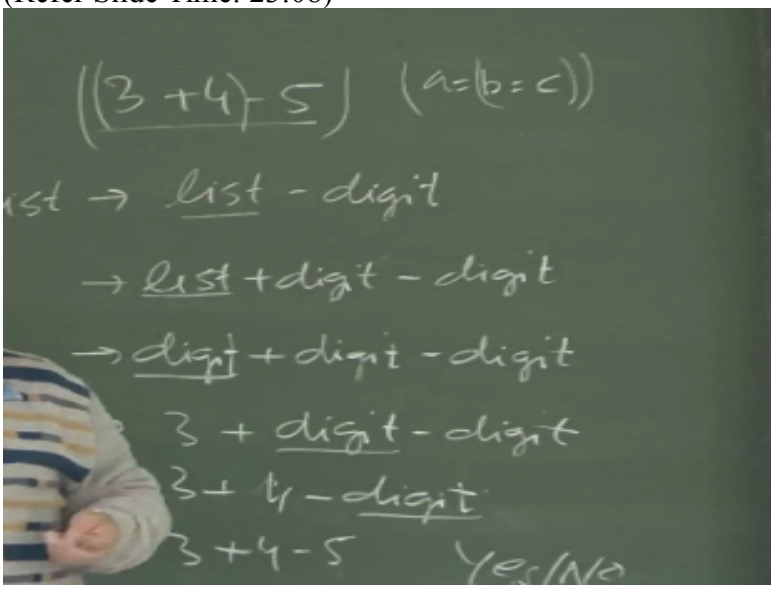
And how a constructed if I have a production A goes to x1,x2,xn and I must have some sub-tree here were the spirits so this is like if I say that if I just look at this part of parse is level by list all the children able by all the symbol on right hand side this is common many of you living peacefully here slide are sleeping and here is one guy resourcing another guy next to you I peacefully sleeping if you feel very sleepy you should use night I will just sleep in one them will in face book till 5 clock in the morning.
(Refer Slide Time: 22:30)



Alright here is example of parser so just expected so is this kind this is you get 9-5+2 what is step for following. Now let us step to agree now what is the maximum that can I constructing

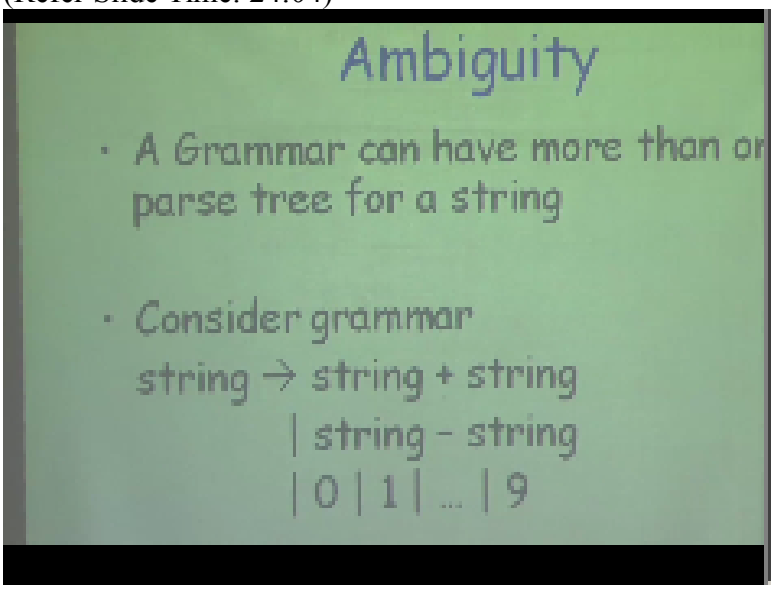
past link in this case which will happen that this is only past here now me change this grammar slide and keep this change string let me say that my grammar something like this suppose I had grammar.

(Refer Slide Time: 23:08)



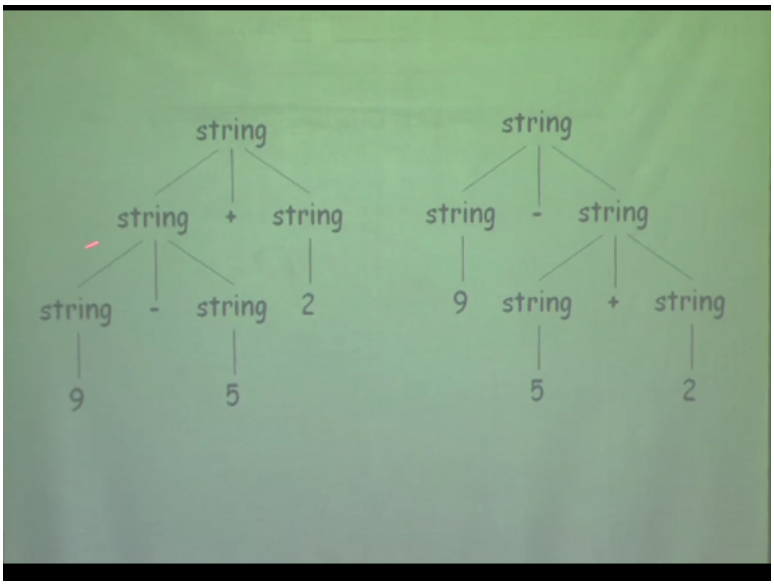
Says list post to list + list or list goes to finals this and then list goes to began with the grammar as suppose to earlier grammar and then I say that now digit goes to know what to have interesting can happen because I was say that I Am looking at 3+4-5 I too have because I can always say that so what will happen here is grammar .

(Refer Slide Time: 24:04)



So I have string .string +string string for on the digit step here now if I try to construct parse.

(Refer Slide Time: 24:20)



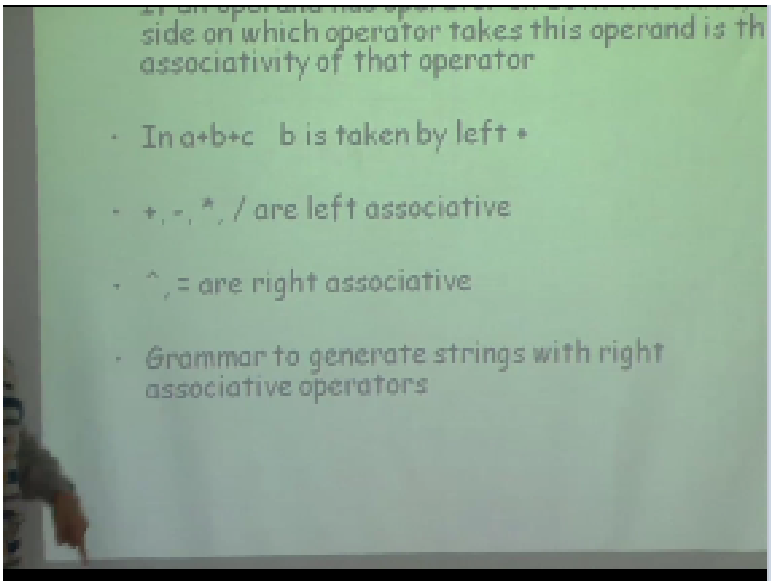
Before this the one parse tree will look something like this that is a string goes to screen per string and then I may decide to expand the leftmost non-terminal by say this will stream was posting - thing and then this was 9 is supposed to five t and this goes to time doing a leftmost derivation. Now I am going to another that post-renovation to be right the same thing from this down so why that post elevation maybe that I say string goes to string - string so instead of using this rule.

I fix up this the first so randomly, I am choosing a rule so I just picked up another rule but I have at least said that I want to variations ,so I not expect the right-hand one but I will expand this and I have to say that this expects tonight and then I decide that I want to now extend this and I expand this by saying that I am using and then again this is my leftmost on top of another so I decide to expand this and now this is the only that permanent so I get to parse trees cool as post elevations.

Which are giving you the same space and this I say is 90 plus rounds so the way this grammar has been written is given rise to an MVP now if I am looking at the process of parsing process of parsing in the language or not and that turns out that does not matter which set of derivations are used I am getting the same answer yes this is the very same so where is the power is ambiguity at therefore a problem so I am always with both the cases .I am getting the same answer these would have a bigger problem you can one case the parsing would not fit that did not happen.

So where is the catch so suppose. I just take this is an issue either of this parse tree or this part both are fine not exactly we will remember the barber school from your plastics mathematics classes when I start with anything this at some point of time I have for dinner and then I am saying this is I will say that I start generating port force of peace and then I pick up core

generating code for this sub tree this will say that I want to subtract five from nine and then we say that whereas when I start generating port for here it will say that I want to add five in two and then I subtract.
Which possibly gives me the correct result first one the second one is obviously and therefore I do not want to generate parse trees which will interpret my expression differently this one actually is meeting with these things then I find that because of ambiguity I can get more than one knee and some of the meanings so I want to make sure that ambiguity is handled in a manner that I never generate more than one positive because there is no question of and therefore.
If you now see that the form of the grammar is important okay the way I wrote this number versus this llama you will find that in this grammar. I will not be able to get it and because please but in this grammar. I ended it and we goes please by doing that most derivation yeah so everybody is a problem because meaning of the program sweeper and that is why one point can be routine and it can be handled in more than one ways so one way to handle any ambiguities that I start looking at all my operators and say that associatively they have certain residences and those must never be valued and the second thing is second way of doing is that I need grammar.
Rather than so if you remember your rank exercise and somewhere in the top you had symbols like this okay and then I could write we remember this is saying that one of the operators winter left to serve you t go to the operators which are associative what are the operators which have five resilience what are the operators which are incidents and so on so you can specify this but this is not part of the downright.
This is what is spoon providing what we would like to do that is that rather than handling it through a tool we make it part of the town so what we take the grammar I will depend upon the tool and their most general techniques for handling this I occupied specifications and make sure that I handle like we will dis correctly and his deputies at this point of time we have no angle at such a help it not be possible which can automatically convert if I give you like this there is no manual to understand how do I write grammars and then actually write grammars.
Which are going to be under use so another aspect so another aspect we come up and we want to handle that the associate activity part of my grammar?
(Refer Slide Time: 31:00)



So if you have an offering and the signs then I want to pick up which operator is going to take this particular operand so you know that when you talk about associatively really say that addition is left associative or addition associative so if B has plus on both the sides we know that I am going to evaluate a plus B first before I add C to this okay so this is what associatively is and we know that all the and exponentiation and assignment top right associative and this way I have it in this grammar you can see that this graph one actually is electrical signal and let because it means that if I look at the first symbol what symbol is same as the left hand side.

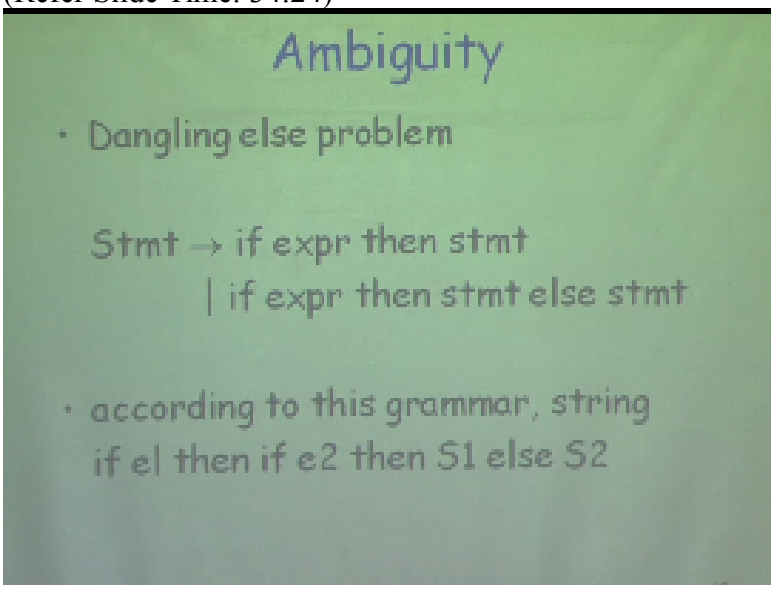
Okay so I can keep on doing a recursion on this and keep on expanding this so I can also generate for these kind of symbols I can write grammars which are right because it and which will enforce right associated with it so if I say assignment then assignment must be expended over something like this so another way to look at this is the cowboy code fragments okay , so mentally if I say that I want to put brackets on this I am going to put brackets like this.

And if I say a is the fine B is the science team. I am going to put brackets I am going to put that it's like this right so basically this is making my rewrite heavy and this is making my pre-lecture okay and the same thing is captured here that when I have write the cursory rules. I am saying right is going to letter being assigned right or letter and then once again coming back to this form that form of the grammar is important and I want to make sure that what we say then all the all that we do this character right we do now and you do not have to specify anything in yeah okay so precedence okay.

(Refer Slide Time: 33:16)

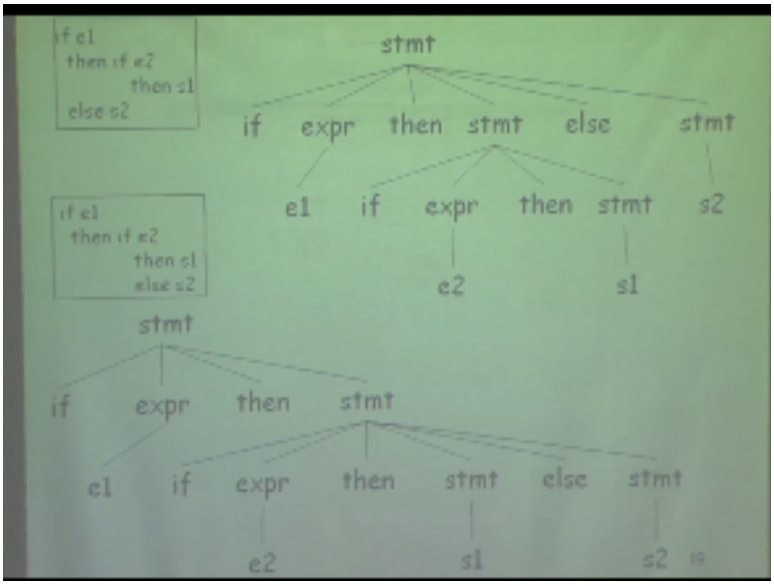
Once again okay that even if I do not put brackets I know because of my knowledge of that so same thing I do with rest of the constructs that although I can have multiple passages for this was

suppose my why and rich my rules by saying that , I have one more rule which says this goes rule is start yes and one more rule is divided by list and then try to pass this thing I can get multiple interpretations of this so one possible interpretation could be this interpretation another interpretation for this as far as this grammar dispenser and you know that one of them is incorrect the first one is obviously not correct.
Because we know that but typically patient a five presidents and therefore this will get evaluated first before the addition happens so I have to make sure that that part is captured so precedence is going to determine the other language constructs okay.
(Refer Slide Time: 34:24)



So there is another problem and this is what we know as thank you girls for that programming languages and therefore in the sentence you will not have equal number of than analysis and certain immediately comes is then what is the association between else's in this else associated which then it is all that question so the grammar is something like this which says a statement space if expression this is the Boolean expressions then statement or if expression then statement and speak and this immediately gives rise to the question which says that I can have any amount of nesting off if there are speakers.
So now let me give you this particular state which is if even then if we do then E1what is the meaning of this in which order and we will valances so if E 1 is true what would I go if even is true what we get executed if we do but if even is false then but why not has to get executed according to this hey where does it say that if this is false then the thing should get executed because I can interpret this as saying that if e 2 then s 1 that is 1 statement and then if even then this is true else. So basically what can happen is if I can have two parts.

(Refer Slide Time: 36:17)

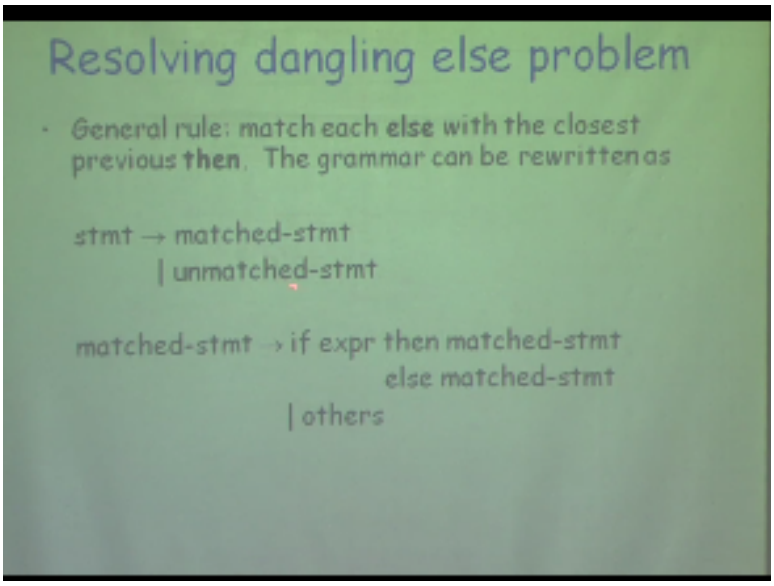


For this is one possible parse tree he says statement goes to if expression then statement l statement and this statement gets expanded to a ten-part and the second pastries of this form it says if expression then the statement and the statement s as so basically question are to answer is whether this else is part of this statement or this else is part of this tip both are valid positives as well as this interpretation is concerned yeah but if I just try to make a brown diagram for this and give you proper indentation.

This will block diagram corresponding to this and this is block diagram corresponding to this and which one is correct second one why is that so what is the rule I am using there in case of metric. I was using mask nothing so all we need to say here is one way to look at that is that he tells must get associated with the school that is them if I just do that that is the resolution now one way to handle that is that either I again now using yak how do I specify that that S must be associated with the truth is that here.

I was able to say what is associated with me what was the Precedence envoy specified that you specified in there so I said there are two ways of realizing grammar one was that I give all this information about associate beauty and residences as part of the truth and second possibly like the grammar research so if I use the first technique and try to rewrite this grammar so that all the associated with even presidencies are taking care. I will do that well at least I am but what we can certainly do is we can rewrite the grammar.

(Refer Slide Time: 38:38)

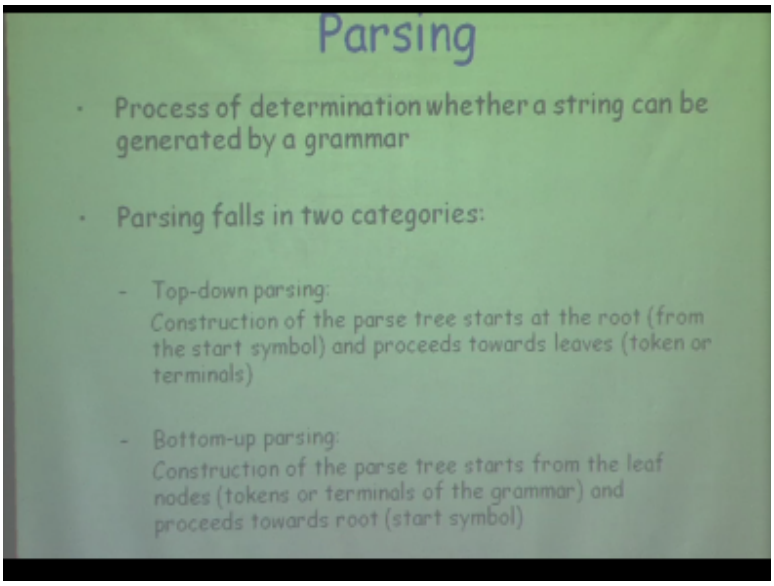


So general rule is so easy enough to be like the grammar so I will show you more than one way love rewriting it so general rule is that match else with the closest previous then and this grammar can be now rewritten by saying that I can I do not have all the balanced set of else's invents so I can either have statements where number of else's are number of associated with each realism so these are all as bad statement so much statement have equal number of X's indents or I can have one match statement where the number of I have more assistant thanks okay.

So those are all unbalanced okay and now if it is matched then everything has to be matched right so if it is matched then if it is if expression then this part needs to be exactly matched because if this is unmatched then this whole thing will become unmatched right and if I say that this part is must have else and this part also must be matched so now you can see that if I am talking about match statement I have equal number of them analysis and then in the next thing also I am only the massive.

So this is make sure that I do not have an equal number of them analysis but if I have others and what are these others which are any other statement other than if that will not change my number of damage but if I have unmatched so first thing that will happen is that if I do not here then I can put anything it does not matter what statement whether this is matched or unmatched okay this itself just by having a time here I am not having an affair but if I say that I have to have this else are shorted associating with this then it means the statement after that must be matched right then else this associated because if this is not matched in then we cannot get associated you get associated with one of their dances this is right.

So now you can see that I am not really using any of the pool specification but I am reading right now by recognizing the fact that I can have more number of dense than else's and then I can rewrite my gravel in this form here I say this is unmatched and obviously this partners could be unmatched but at least this part has become matched and if you have completely so this is one way of writing is not the only way of rewriting and I will show you more ways of rewriting.
Others all that is any statement so for example assignment why people do anything which is other than if they meant this one so I am saying that I am now trying to write specifications of statements which have unequal number of seminal now if it is unequal number of them analysis and I also want to capture the fact that each else must associate with the flow of them then if I look at this if I have lift the else then to make sure that this else associates with them okay.
Between this I must have equal number of then and else's then only you can have but the whole thing has to be unbalanced then how do I make whole thing unbalanced by having an unmatched yeah right okay so my name is on the question yet switching here right then the problem will be I will not know how to match dissonance because this part is unmatched that means it has more than then this else will get associated with another then and else.
No unmatched is not valid unmatched is okay but valid interpretation is that else was associated with some one of the tens but if I am writing a rule like this then this in this rule this is the right matching right so this is what I am caption this expansion or the positive for this is saying that this is one statement right so in this statement I want to make sure that this expansion happens such that this else in that mattress.
So basically what you have to remember is that in this particular grammar rule if this is my expansion this is one statement if expression then matched else and that way if I want to have this association then I must make sure that between their nails have a great night and this will make sure that if I write my so let us move on to parsing now.
(Refer Slide Time: 44:47)



And see that with all this structure of the novel and some of the transformations of the grammar I will live for later are when we little more about parsing so what I want to do here is to begin with right certain grammar have certain strings and check whether these strings belong to the language specified by the doubt it so this is the process of determination whether parsing belongs to the language which is the generated by a grammar.

Parsing is following in two categories one is top down parser the another is bottom up parser okay, and now you find these things with respect to the syntax in parse tree I just constructed we say that if construction of parse tree starts from the roof and goes up to the leaf nodes this is calm manner in the construction of the path we start from the leaf nodes and then finally the root gets constructed let me say that this is really bottom of the parses.

So one thing you think or another thing you can do is that if I am doing this derivation I am saying that in case I start deriving from the start symbol and then I finally reach a string okay then I say this is really a top down parcel because this is where I started constructing or if I start from the leaf nodes and slowly keep going back and say that yes I can reduce this whole string to the start symbol then that is the bottom of parses okay because a parse tree which is getting constructed from the leaf nodes finally it will say whether I can construct a route note corresponding that and all.

So these are techniques of parsing we use and enter is a tool which supports top down parsing and yeah inviting other tools with support parses we also see methods of doing top-down parsing versus bottom-up and we actually see that bottom of parsers are more powerful compared to top down parser and when I say more powerful what that means is that bottom of passes we will be able to handle more languages then what a top-down parsing ca n handle if the set of languages.

which can be specified by a top-down parser or can be handled by a top down parsers fewer than bottom of parsers and therefore although we use both the parsers ultimately little more time bottom of the parsers so this is where I want to break today and we continue our discussion the next class from this one the next class is on Sunday 11:45.

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