Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 27
Dependency Grammars and Parsing-Introduction

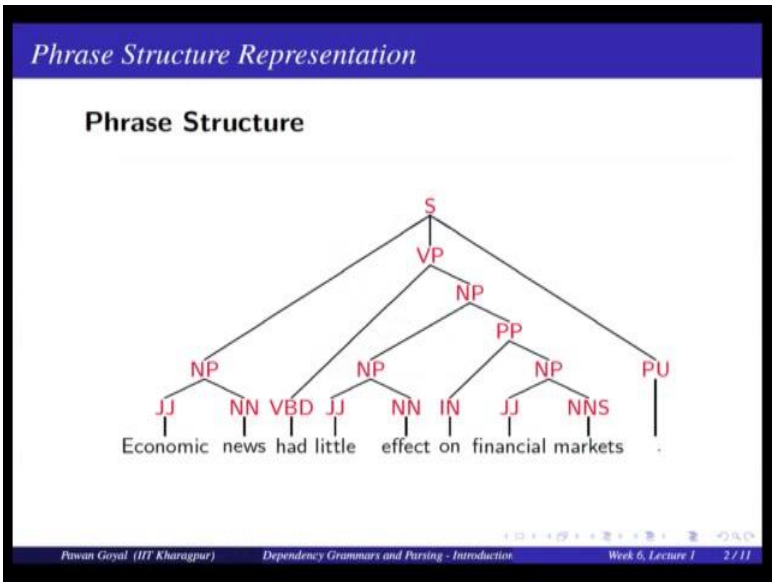
Welcome back for the sixth week of this course. So, we had already started our discussion on what is parsing, so this is the topic in syntax. And we talked about a constituency parsing approach by using the probabilistic context free grammars.

Now, we will use a slight different version of parsing is called dependency parsing. So, where instead of what we did in the case of constituency parsing where we were finding out what are the different word groups in terms of noun phrases, adjective phrases, verb phrases, we will directly find out what is the relation between any two words in a sentence. So, dependency parsing notion has picked up a lot in research field in the last few years, but as such it is origin is very very old, many many different linguistic traditions.

For example, for the Sanskrit the famous Sanskrit grammar [FL] that are the dependency relation between verb and different words in the sentences. So for example, if I have a sentence like Ram eats apple a simple sentence I will see the verb eat is the main verb here and what is the relation of Ram, so Ram is subject of this verb and apple becomes an object of this verb. So, like that I am here I am trying to directly say in this sentence if I am in different words are these related somehow and what is the relation between them. So, this is my dependency structure.

In this lecture we will see what is the formal way in which we can define dependency structure, what are different linguistic constraints, etcetera that we need to impose in this structure. And then we will start talking about the some of the models that are very popular for getting the dependency structure it is starting from the sentence.

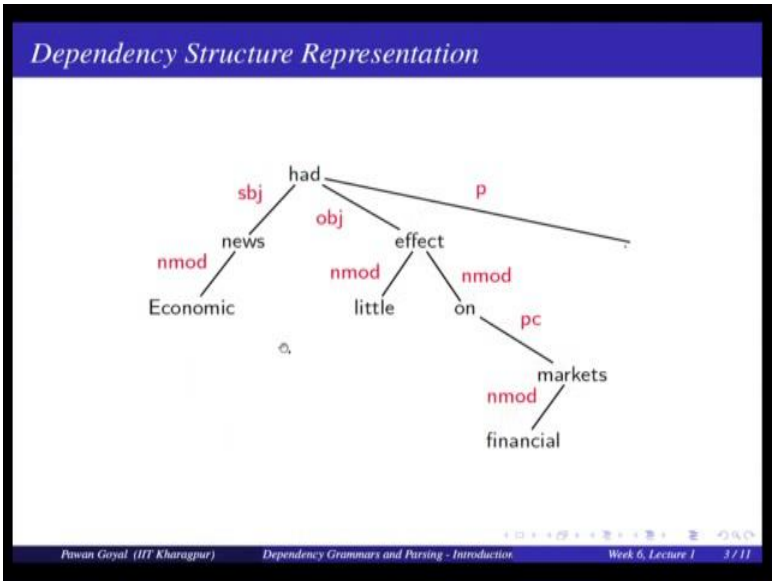
(Refer Slide Time: 02:22)



So, this kind of tree we have seen in the case of phrase structure, also called the constituency structure. So, what we have seen here; I have sentence economic news had little effect on financial markets. So, they are different group of words (Refer Time: 02:36) as single unit; so economic news make a noun phrase.

Similarly this whole thing makes a verb phrase and they together make a sentence. So, like that what we have seen here which words are grouped together, and how this whole sentence has been arranged; that is what you see in a phrase structure.

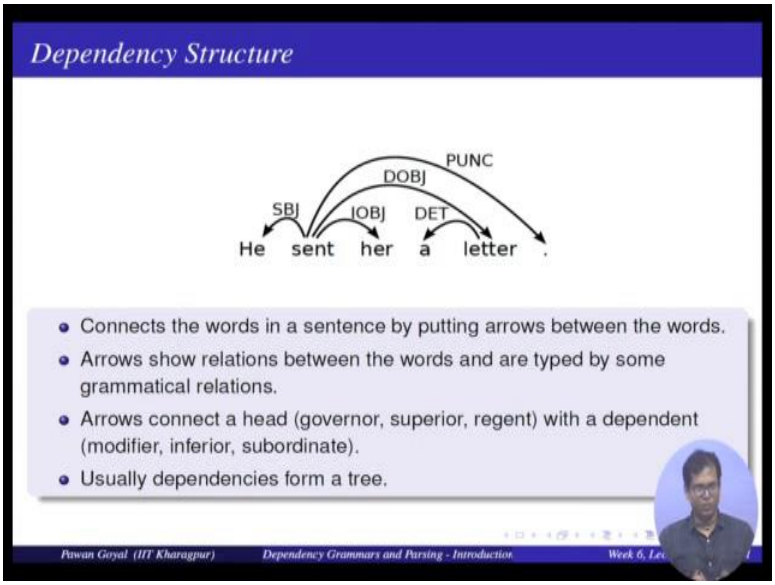
(Refer Slide Time: 03:01)



Now, how dependency structure different. For the same sentence let us look at the dependency structure. Here what are you seeing; now this also looks like a tree, but the nodes here are the words themselves not the phrases. And different words has been have been connected by some edges and they label by some relation like, had and news are connected and the relation is subject. Saying that news is a subject for the verb had.

Similarly, economic and news are related. So, this is economic is a noun modifier for the word news. And so on lot of different words are connected in this sentence. So, in general what we are seeing two different words in a sentence are connected by certain relation, and this is a directed sort of relation. So, economic and news are related by saying economic is a noun modifier for news.

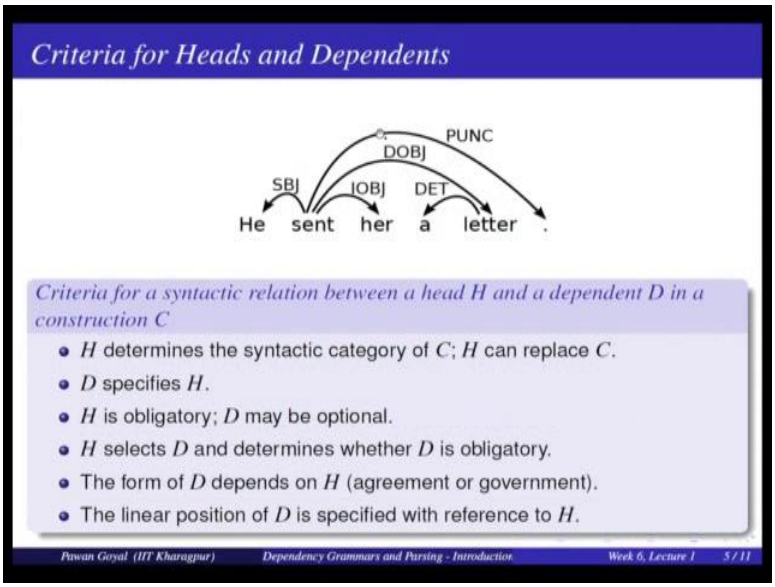
(Refer Slide Time: 03:56)



In general, how we can define a dependency structure? So, we take another sentence here ‘he sent her a letter’. So, what you are seeing here? First you are seeing some 5-6 words or if you take the punctuation. So, they are 5 words as such ‘he send her a letter’ 5 individual words. So, what do I do in dependency structure? So, we are connecting the word in the sentence by putting arrows between the words. So, I am putting an arrow between sent and he and these two are connected. So, ‘he’ is a subject for sent. Similarly I am putting an arrow between sent and her. So, in that ‘her’ is in indirect object for ‘sent’.

Now, what do these arrow show? Which arrows are showing means the relations between words and also typed by some grammatical relation. So, they are saying these some relation and what is the relation, it is the subject relation; 'he' is a subject for 'sent'. And what do these arrows connect? They connect a head word like sent with a dependent word like he. And there are some in linguistic there are some other terms like head can be called governor superior regent and the dependent can also be called a modifier, inferior or subordinate. Usually, as you would see the dependency which will form a sort of tree structure.

(Refer Slide Time: 05:19)



So, what is important here? So, dependencies you have a head that connects to a dependent by an arrow, so arrow points from a head to a dependent. So now, are they said some formal criteria for saying in a sentence if two words are connected what will be a head and what will be the dependent. So, there are some (Refer Time: 05:39) they may not be applying everywhere, but may be applying it certain points. So, you will see some say examples for some of these.

So, what are these criteria for? Defining a relation between head and dependent in a construction; so first criteria you can see is that the head determines what is the syntactic category of the dependent or sorry, what is syntactic category of the whole construction. So, like if I see this construction 'a letter' the whole syntactic category governs simply by the word letter. And that is why the head itself; so the word letter becomes the head.

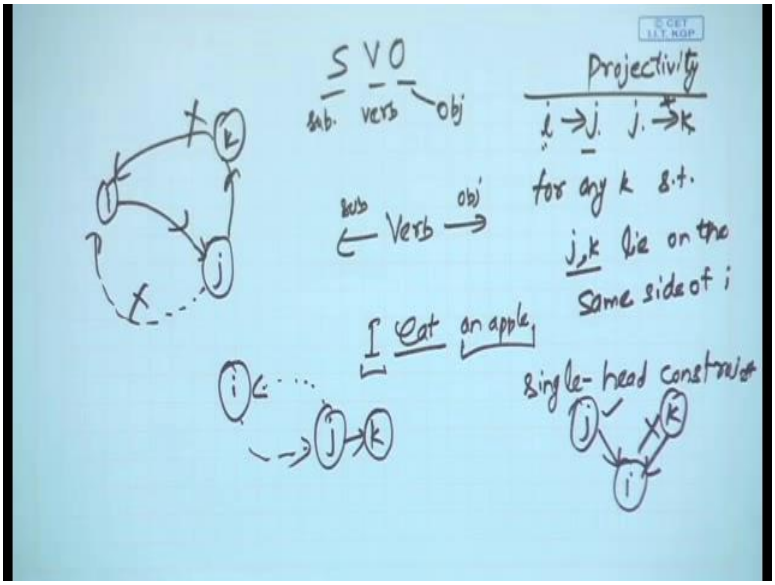
And this word letter can also replace the whole construction C. So, in place of the whole thing a letter I may also write simply letter.

Now D specifies H; that means D is giving further specific information about H. So, if I say only letter may not be as specific as saying a letter. So, all these determine they give some additional information. Similarly here, so sent is the head and letter is the modifier. So, if I just simply say he sent it is not very specific. To whom did he sent, so there I have to specific particular modifier; he sent her. Similarly, what did he sent. So, there I have to put the word letter. So, what you are seeing the dependence is further specifying my head. Now in some cases; so head is always obligatory I need to put head like letter is necessary to put, but D I may put it optional in some cases. So, I may also say he sent her a letter, so D the dependent may be optional in certain cases.

Now, the word the head word it selects my dependent D and it also determines further the dependent is obligatory or not. And in some cases the grammatical form the D takes will also dependent on my head. So, this is also called the agreement or government; government in the case of linguistics. So, like the form that use for your verb in your construction will be similar to the form of the subject and object that we are using.

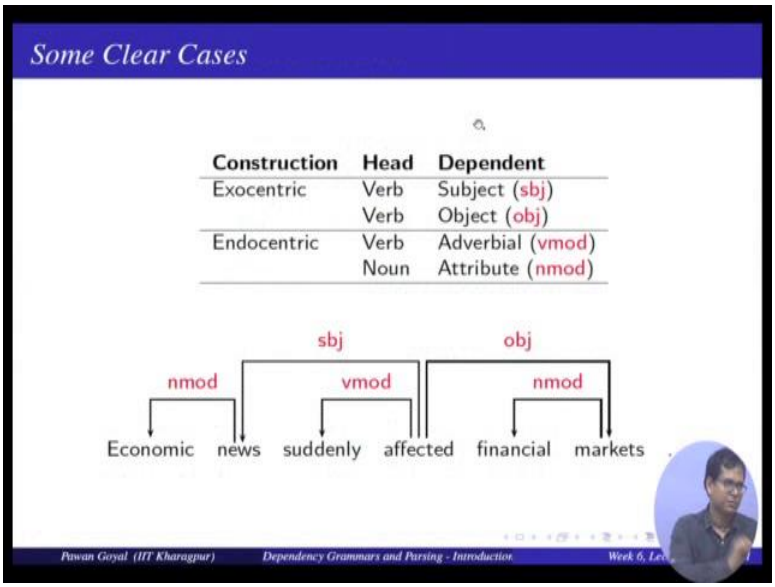
Also with the linear position of the dependent is specified with respect to the head H.

(Refer Slide Time: 08:25)



For example; so English follows this as we are constructed right. So, you get the subject first, then the verb, then the object. So, what we have seen? The linear position of the dependent is determined with respect to the head. So, if I know the verb has come, so I know subject will come to the left and object will come to the right. Like I want to say sentence where I am saying I am eating something. Who is eating the subject will come on the left, I eat and what do I eat say an apple that will come on the right. So, the position of the dependent it is specified with respect to the head, whether it is to the left or to the right.

(Refer Slide Time: 09:09)



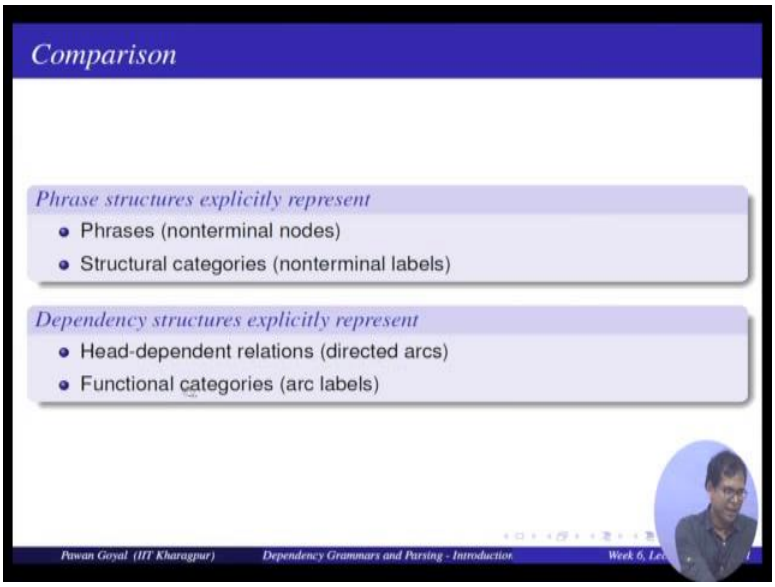
Now, there are some cases that are clear where you can easily find the dependencies, so for example if I look at the exocentric and endocentric constructions. What do I mean by endocentric and exocentric? So, endocentric construction is one where, one of the entity here can actually fulfill the whole grammatical function of the complete construction.

So, if you look at this verb and adverbial relation verb modifier, what is the example? Suddenly affected; this may, doing some grammatical function suddenly affected and why it is endocentric because only one word here like affected can fulfill the whole grammatical function for the whole construction suddenly affected. Suddenly is simply modifying that. Same way for non modifier like financial markets, so markets can fulfill the grammatical function for whole thing financial markets. It is not the case for exocentric construction like verb and subject.

So, if I take affect and the news. Affected cannot fulfill the whole grammatical function for news an affected, I need to have the word news there. So, this is called exocentric. Endocentric, one word can fulfill the whole function exocentric a single word cannot. But the cases of exocentric what will become the head and what will become the dependent. Like if I have the verb and subject I know that verb will become the head and subject will become the dependent. Same way verb and object; verb becomes the head and object becomes the dependent.

In the case of endocentric again, the particular word that can fulfill the whole function can become the head and the other word can become the dependent. So, here verb is the head affected and suddenly becomes the dependent. Similarly, here market is the head and financial becomes the dependent.

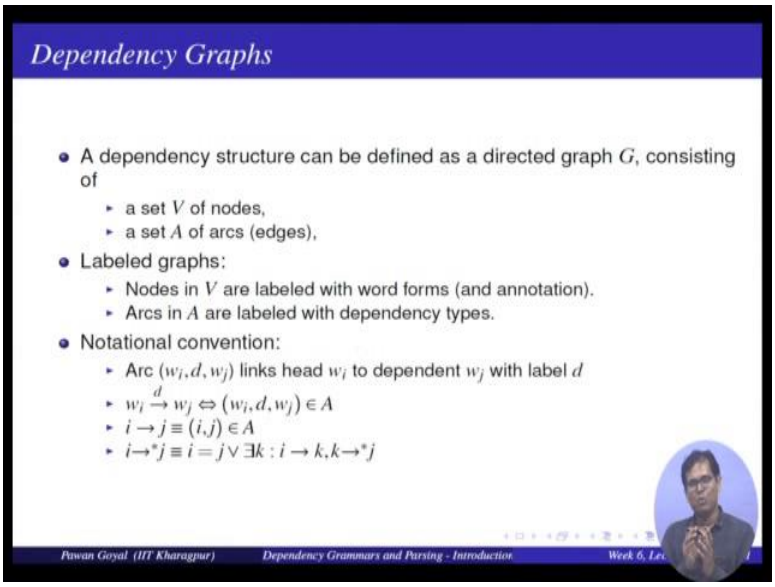
(Refer Slide Time: 11:11)



Now, if you just try to compare what is the difference between a phrase structure a representation and dependency representation. What we see is that in phrase structure we have very explicitly denoting what are the phrases, these non terminal nodes and labels. In the case of dependency relations what are the explicit denoting? They are words, but what are the relation between different words and what is the grammatical category of each individual relation? So, in the case of phrase structure I denote what are phrases and what are the structural categories; like noun phrases, verb phrases and so on.

In the case of dependency structure what do I represent, what are the various head dependent relations like in the case of directed arcs? And what are the functional categories that are; what are the different arc labels that I am giving to various arcs in my dependent relation?

(Refer Slide Time: 12:07)



Now, formally can I define what is dependency graph? So, I have to define it for the words in my sentence. Formally I can say that the dependency structure can be defined as a directed graph consisting of a set V of nodes and a set of A of arcs. Now this is a very generic way in which you define a graph. In a graph you have a set of nodes and set of edges that are connected different nodes. Now is there something more than this (Refer Time: 12:40) in the case of dependency structure.

Now in the case of dependency structure; so this is the labeled graphs, what are the nodes in my graph? The nodes in the graph denote the word forms, the word that I that I am counting in my sentence. And with that I might also have some annotations like what is the part of speech category of these words. (Refer Time: 13:04) I can also put in my nodes. And what are the edges they are labeled edges with the dependence relations. What is the dependency time if it is subject, object, noun, modifier, verb, modifier etcetera?

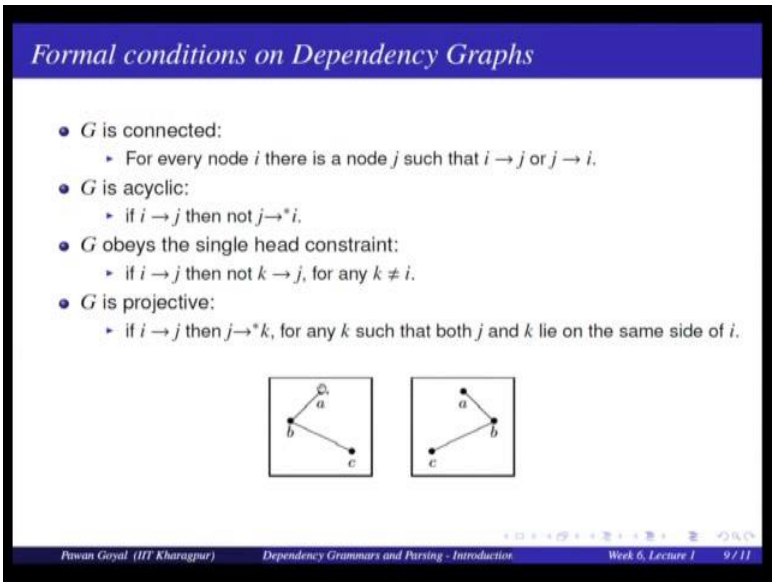
And some simple notations that we used for this is that if I am saying that word w i is connected to word w j with dependency relation d. I can use this arc w i, d, w j; it says

that w i is the head, w j is the dependent and d is the relation. So, I can say eat - subject or he there is another way in which I can note the arcs by saying this is the headword this is the dependent word I am denoting an arc from a head to dependent word and putting a label on top of that this is another way. And this is equivalent to saying w i, d, w j are in the set of arcs.

There are some other conditions like; if I say that from i this relation i i to j direct relation then I can write that i j in my set of arcs. So, I am not writing the dependency relation here, but I am saying i and j are in the set of my dependency relations. Similarly, I can do a closure of this saying that from I there is a there is path to j; that means in any number of steps I can go from i to j. If and only if either i and j's are equal or there is some case such that I go from i to k and from k you can go to j any number of cases. This is very standard way in which you define the closure relations.

So, with the dependency labels I can define a single dependency or I can also define a closure of that. So, I hope this notation is clear. Once we have this notation what does the total form of conditions that we can put on the dependency graph. On the graphs so that the denoted dependency graph for a sentence, what are the formal conditions?

(Refer Slide Time: 15:24)



So, what are the formal conditions? First condition is that the dependency graph that G that I get is connected. By connected what do I mean, there is no node in my graph that is isolated from the whole graph. Or in other words if I take any node in my graph I can

always find another nodes such that either that node has an incoming link from that previous node or an outgoing into that node. So, it is connected to some of the node in the graph attached it is not isolated. So, this is the simple condition for saying when the graph is connected.

The second condition is that my graph is acyclic; there is no cycle in the graph. So, what do I mean by that? If I say that from a node i I have a label, I have a path or I have a dependency from i to j. There is no way I can have a path from j to i. So, there is no way I can go back to I, if I am going from i to j I cannot go from j to i either directly or by following something k and this, so this is not allowed. So, because this will make a cyclic i to j, j to k, k back to i; so cycles are not allowed in the case of dependency graph.

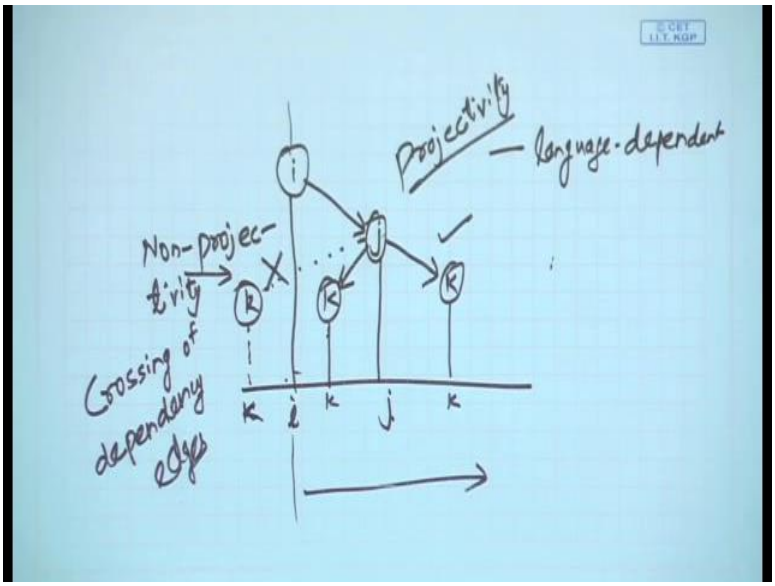
Similarly, if I can show the previous one that means that if I take a node i in my graph it should not happen that and they are connected and i is isolated. I can always find some nodes such that either there is an incoming edge or there is an outgoing edge one of these, they cannot be both because of this principle of a (Refer Time: 17:18).

What is the third one? Third condition says that my dependency graph g obeys the single head constraint. So, what is single head constraint? So, single head constraint means if I take a node i in my graph they can be at most one head for that. So, if I say that j is the head for I; so j is the head for i I cannot find another k so the k is also head for i. This is not possible, there will be only one head for i node at most is called the Single Head Constraint.

And then finally, we have the fourth constraint that is g is projective. It says if from i I can derive j, and from j so I am using the word derive here derive means from i, i is the head and I have the dependent j. Then from j I have a dependent by some number of edges k, for any k such that j k lie on the same side of i. And this is the Projectivity Constraint.

And what do we mean by that? If I were edge from i to j and from j I can have in any number of steps I going to k, then j and k should lie on the same side of i.

(Refer Slide Time: 19:15)



So, that is if I am saying this is my i and from here I have this relation to node j. Let me just project them I is here and j is here. Then what I am saying is that if j connects to any dependent either directly or indirectly that k has to be on the right hand side only. So, there are two possibilities suppose I am taking direct connection I can connect to k this is allowed or I can connect to a k here. So, these denote the linear order in the way they are occurring in the sentence.

So, i and j are here, so k can occur either here or here or anywhere else in the right. But, what is not allowed is if j is related to some k here that is not allowed, because now j and k are on the different side of i. So, whatever side j is k also should be on the same side. So, if this situation happens this is called Non-Projectivity. Also there is another term for that it is called crossing of dependency edges. So, you are seeing here this line and this line are crossing here if I am taking this particular constraint. While they do not cross if I take this constraint or this constraint and you can see if you can go any number of steps they will not cross and this particular constraint is called the Projectivity Constraint.

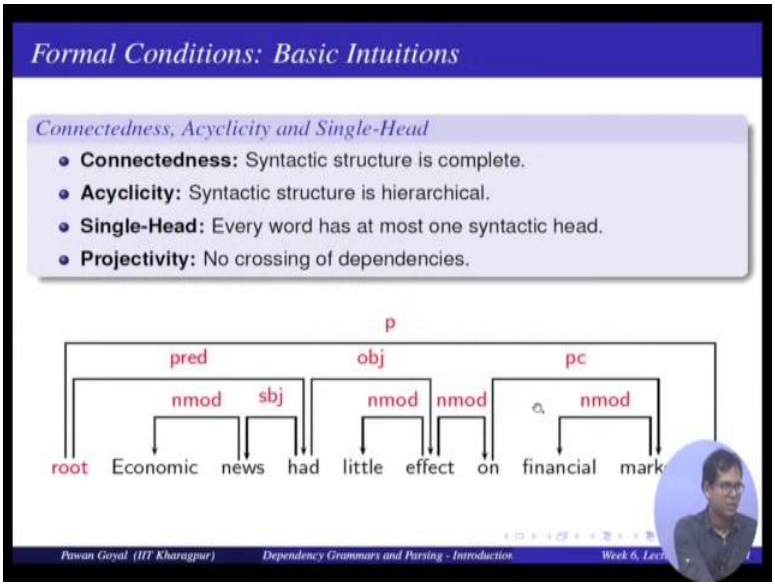
Now, what is important is that this constraint of projectivity is a many times dependent on the particular language that you are choosing for your dependency construction. So, it might happen that there are certain languages that do not follow this projectivity constraint and some languages like English they very they quite regularly follow this projectivity constraint. So, this is like language dependent. So, the methods that

dependency parsing that we will study in this week, so we will study one method that that needs projectivity constraint another method that does not need this constraint.

So, but these four constraints are very very important so that you can come up with the good algorithm for a dependency parsing of the sentence. So, this is an example of what is projectivity. So, if I have a link from a to b I cannot have a link from b to c, because c is on the other side of a.

Similarly if I have a link from a to b I cannot have a link from b to c, because they are crossing.

(Refer Slide Time: 22:23)

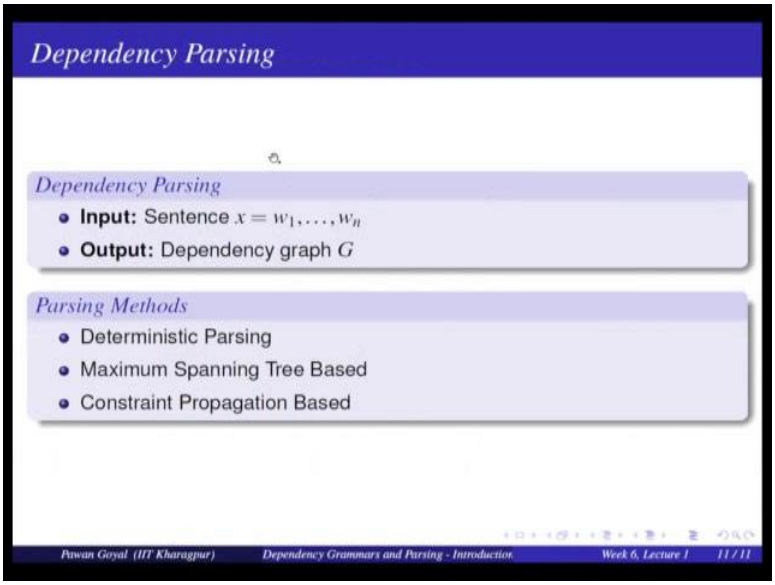


In general, what do they mean these four conditions? So, connectedness means that the syntactic structure obtaining my dependency graph is complete. I am not having any isolated nodes that are not connected to my whole structure. In condition of acyclicity means that the structure is hierarchical there is no cycles inside.

What is the single head constraint? It says that every word has at most one syntactic head it cannot have more than one syntactic head. So, in other words we are saying it is determined only by one word not by multiple words. Remember we were saying that the linear position of a dependent sometimes depends on the head. So, node 1 2 different heads to decide the position of the single word and then they might be conflicts the same

word has only one head that determines it is syntactic category and in other things. There is no crossing of dependencies that is the last constraint of projectivity that we also saw.

(Refer Slide Time: 23:34)



So, now once we have these four constraints what is my dependency parsing, so this is how I can define the dependency parsing. I am given a sentence x that contains words w 1 to w n and I want to obtained a output as dependency graph, given this input of sequence of words in my sentence I want to obtain my dependency graph that follows that constraint that we saw earlier. So, it is like single head constraint and projectivity if I am imposing that and all that. So, I want to obtain dependency graph.

So, what are different methods that that we will be using for this? We will talk about different methods so like one method will be deterministic parsing this is a very very popular method. Then a method based on maximum spanning tree. And finally, you can also do it in a constraint propagation method. So, in this course we will talk about the first two methods and how you can use that by having some sort of labeled data and doing some learning from there. So, it is like a data driven parsing. And the final method is something that you will use when if node have any labeled data, but you know what are some of the constraints that your grammars follows. So, can you encode those constraints inside your dependency grammar and obtain dependency graph for a given sentence.

So, this is good for the languages where if node have a labeled data, but you have the labeled data in terms of dependency graphs you probably go for a data driven approach. So, starting from the next lesson we will start talking about these driven approaches for dependency parsing.

Thank you.