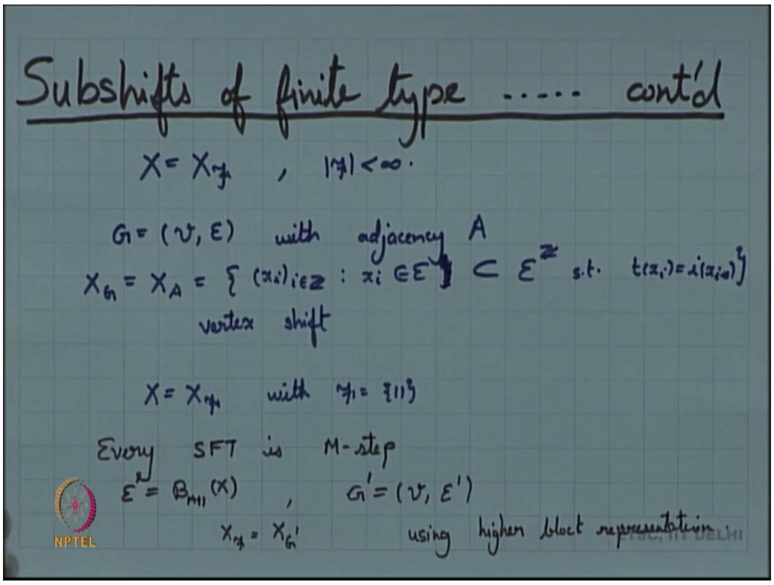
Chaotic Dynamical Systems

Prof. Anima Nagar
Department of Mathematics
Indian Institute of Technology, Delhi

Lecture - 26
Subshifts of Finite Type... Cont'd

Welcome to students initiated the study of sub shifts of finite type. Basically sub shift of finite type are those shifts for which the forbidden blocks can be represented finitely, and today we shall again continue with the same thing. But let me recall a few things that may we may need which we did last time. So, first of all, I start with the definition of sub shift of finite type.

(Refer Slide Time: 00:47)



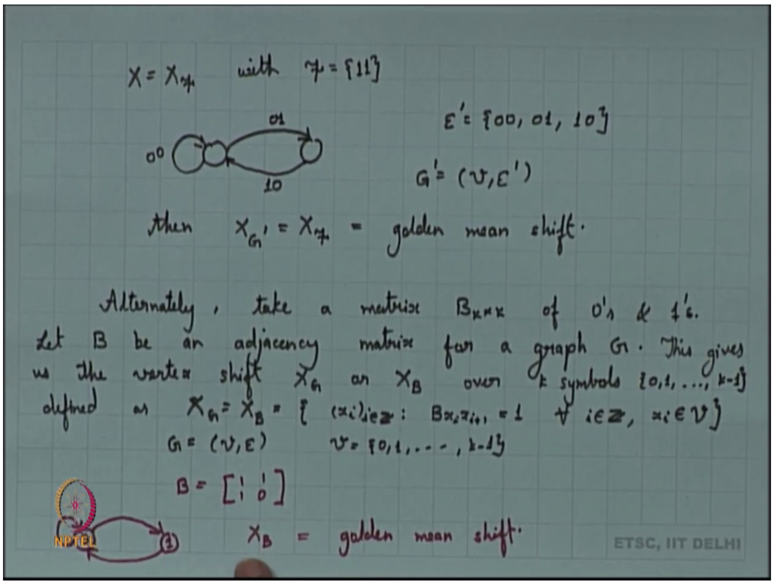
So, sub shift of finite type is basically a shift space for which the forbidden blocks right, the representation of forbidden blocks is finite. And we had again seen something else over here and we had seen that. So, we have a graph G v is the set of vertices and e is the set of edges then for this graph G with adjacency matrix A we can define a shift which we can call it as X G or X A to be the set of all X i. So, basically this is a subset of the set of edges to the power z such that your terminal part of X i is equal to the initial part of X i plus 1.

The initial vertex in the term in the terminal vertex of X is same as the initial vertex of X i plus 1. So, all such factors right all such sequences is something which gives us. So, this is a by infinite walk on the graph G and that gives us something called the vertex shift and we had seen that the every vertex shift is a sub shift of finite type and also that every sub shifts of finite type cannot be represented as a vertex shift. So, we had considered this example that you take X equal to X f with f being just the block of 1 1, then we could not represent this in terms of a vertex shift. But then we are interested in representing every sub shifts of finite type supposing we are interested in representing every sub shifts of final type as a vertex shift. So, what can be done for this particular such particular cases?

So, all we can say here is that, our sub shifts of finite type again this is something again which we are recalling again that every sub shift of finite type is an M step sub shift of finite type right there is some M for which it is an M step sub shift of finite type. And for this sub shift of finite type what we have here is that you consider B M plus 1 of X, which is basically the set of all admissible blocks of X and then all admissible blocks here and then you think of that part that fine look into this as your edge shift. So, give us to be let this be the set of edges and then we can always find another graph G. So, let me call it G prime with some set of vertices and the set of edges as e prime such that your X f can be written in terms of X G prime.

Now we shall try to see how this can be done this is basically done using the hire block representation. Let us take a case of the same example here. So, again we have our sub shift of finite type we have X equal to X f with f being in the block 1 1.

(Refer Slide Time: 04:47)



Now what happens in this case is we can think of again 2 vertices here; our edge 0 0 goes here now we this is a higher block representation. So, one has to be very careful since we have a 0 here, the next part here should start with a 0. So, we have a 0 1 here, and again since we have a 1 here the part here should start with 1. So, we have a 1 0 here. So, now, this is an admissible graph right and one can see that this graph this edge shift right with my edges being now 0 0, 0 1 and 1 0.

With these edges this gives me a edge shift right this gives me an edge shift and that edge shift is basically my this golden mean sub shift of finite type. So, what we got here is that supposing now my G prime happens to be v e prime, then X G prime is same as X f which is basically the golden mean shift. So, one has to get into a higher block version to make to convert a sub shift of finite type into a edge shift and hence we can say that every sub shift of finite type can now be represented as an edge shift right, but what are the alphabets there right, it depends on whether you can directly write it right or you have you could need to take a higher block version and then write it right. So, this is one case the other thing alternately way of looking into sub shift of finite type is that. So, alternately take a matrix B.

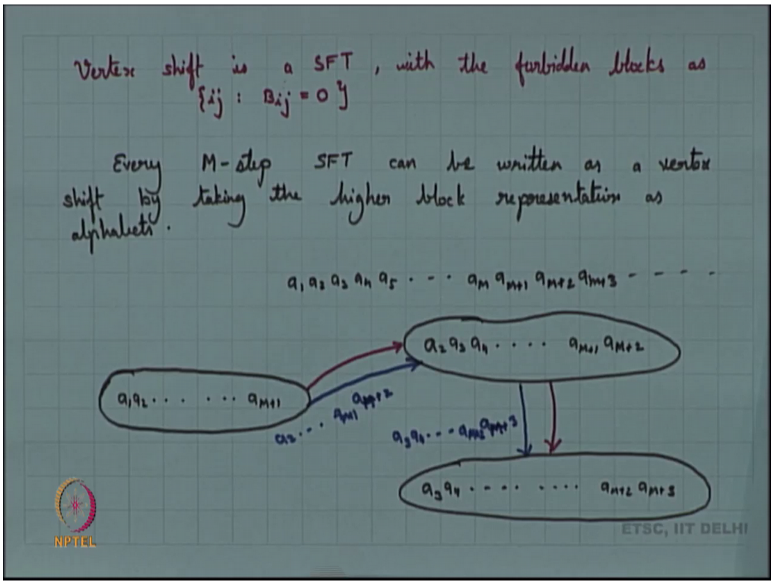
So, let me specify that B is a k cross k matrix. So, take a matrix B of 0s and ones. So, the only entries of B are 0s and ones and let this B, B an adjacency matrix for some graph g. So, let B. So, we can think of this graph G will have. So, this graph G will have k vertices. So, now, we can think of that this gives us the vertex shift which we can call it as X G or we can call it as X B given over the k symbols. So, we can think of the k symbols as 0 1 because k minus one over the k symbols defined as this gives defined as. So, we have this X G this is equal to X B this is basically again I am looking into X i i going I in z such that my B of X I; X i plus 1; this is one.

If it is one I can define the sequence this is permissible for every I belonging to z and my X is basically belong to this vertex v. So, vertex v of G where I can think of G as the set of vertices here, G is v e. So, my v happens to be this particular set 0 1 up to k minus one. So, we think of this vertices and we say that there is an edge between 2 vertex if you have B X I; X i plus 1 to be equal to 1. So, in that case we say that this this vertex becomes. So, X is the vertex; say for example, we can think of our B to be equal to again we take this matrix 1 1, 1 0 and for this we can easily define the vertex shift. So, now, since I have this 2 cross 2 matrix; I have 2 vertices.

I can label them as 0 and 1 and then I want to see that between 0 and 1 between 0 and 0 is there an edge. So, this is one. So, between 0 and 0 there is an edge here between 0 and one also there is an edge here between 1 and 0 there is again an edge here, but there is no edge between 1 to itself right because this is 0 and hence one can think of that that we get here your X B right and if you look into this edge shift this is basically here golden mean shift. So, we get the golden mean shift here looking into it as an vertex shift. So, we can a either represent a sub shift of finite type we can represent the sub shifts of finite type as a vertex shift or you can sub shift represent our sub shift of finite type as an edge shift. So, you have shift and you have vertex shift you have both the cases here

Now, we see that this vertex this is sub shifts of finite type.

(Refer Slide Time: 11:20)



So, our vertex shift is a sub shifts of finite type where you are looking into the forbidden blocks right. So, with the forbidden blocks all i j right such that B i j is equal to 0. So, finitely many of them, this is always going to be a sub shift of finite type; how do we look into an M step sub shift. So, this was just a 1 step sub shifts of finite type how do we look into an M step sub shift of finite type how do we represent that in terms of a vertex shift. So, we note here that every M set M shift every M step sorry.

Sub shift of finite type can be written as a vertex shift by taking the higher block representation as what I as alphabets. So, what you have here is that is supposing. Now if I look into a block; right. So, this sub shift of every sub shift of finite type will be M step. So, supposing I am looking into this block a 1, a 2, a 3, a 4, a 5; right a M a M plus 1 a M plus 2 and so on, then we know that this a to the a up to a plus a M plus 1 this is permissible. So, we start with a vertex here. So, I call this vertex as a 1, a 2, up to a M plus 1, I take another vertex here a 2, a 3, a M plus 2 and then I can take a third vertex here a 3 a 4.

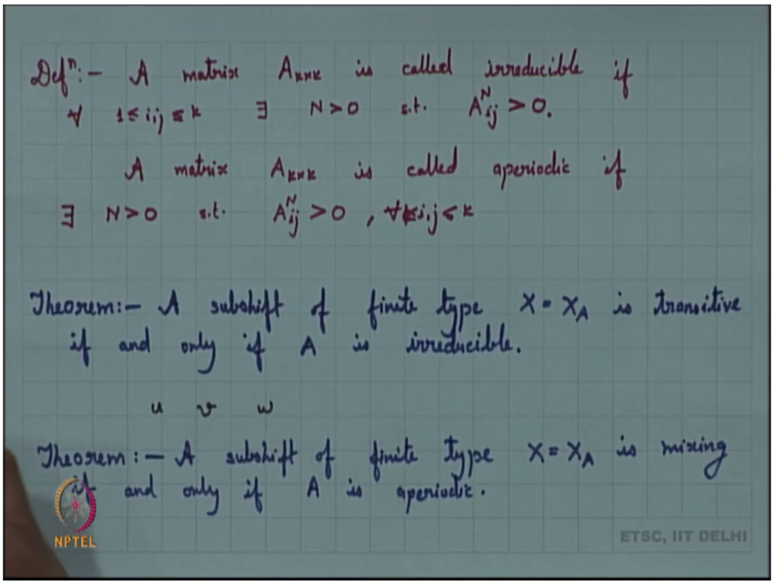
So, I am looking into all M plus 1 blocks here. So, this is one vertex this is another vertex and this is a third vertex. Now when we are constructing a graph, this since this is a higher block representation, we see that this part should coincide with this part and if the part coincides with this part right, then what do you have here is you have an edge here. Now since this part does not coincide with this part right we cannot have an edge from this vertex to this vertex and then when you are looking into this part right. So, you say that fine from here in this part basically this part the last part terminal part here coincides with a term the initial part here and hence there is an edge from this to this part.

So, you are drawing an vertex shift in this particular manner. Now you could also think of this as an edge shift; how do you think of this as an edge shift. So, supposing I have taken these as my vertices then you say that you want to draw edges and your edges should be labelled also. So, what you would try to draw here is you say fine between this and this. So, this becomes this combined with a M plus 2 happens to be a permissible block. So, you would draw a shift here you draw an edge here and you call this edge as a M plus 2. So, you have this factor a M plus 2 takes you to this particular factor. So, then you are removing the first part you are remove your just retaining the last M plus one. So, last plus 1, this come brings you here, then this again is permissible. So, this is like this block coincides with this block.

So, again there can be an edge from here to here and you take an edge from here to here and you call it as a this was M plus 2 this is a M plus 3 and you can have an edge from here to here. So, the higher blocks definitely help us in converting sub shift of finite type or converting a graph of the sub shift of finite type and hence both we can say that whenever we consider any sub shift of finite type like the equivalently have a graph representing it whether we consider it as a vertex shift or we consider it as an edge shift there is a graph representing it and whenever we have a graph we know that there is an adjacency matrix associated with the graph. So, every graph has an adjacency matrix and. So, there is a matrix basically which is responsible or which characterizes.

Every sub shift of finite type, so for us a sub shift of finite type is same as a matrix and now if you want to code it up right it is very easy you are just using a matrix to code a sub shift of finite types with the entire dynamics can be just coded using the matrix. Now what is more important here is to see; what is this characterization; what is this relationship that the whatever what is this information that we can get from the matrix which can help us reduce some kind of results or some kind of properties dynamical properties of a sub shifts of finite type. So, I would just like to recall some definitions of a matrix.

(Refer Slide Time: 17:38)



So, we start with this factor, we look into some definition here. So, a matrix. So, we let me take again k cross k matrix is called irreducible if for every pair. So, I am starting with i and j less than or equal to k, there exist an n right some n which is positive right some natural number such that I take a to the power n and then it what its i jth entry. So, this would be positive. So, we say that a matrix is irreducible if whenever you have any pair i n j. So, you looking into any combination i n j between one and k then there would be some n such that A i j k if you are looking into a to the power n then this its i jth entry is positive. So, that is what is call an irreducible matrix and again we have something more about a matrix.

So, we say that a matrix is called is called a periodic if again; what we have here is we have now something different from here if there exist an n positive such that A N i j is positive right for every sorry between one and k. So, for every i j between one and k if your A N i j is positive then we say that this matrix is a periodic the only difference between the 2 turns out to be that for this particular n it is quite possible that some i prime j prime right the i prime j prime entry could be 0, but for here whenever we take a n right all entries are strictly positive . So, that is basically what we call as an a periodic matrix and now we have a theorem. Now the theorem says that a subscript of finite type now we can we know that a sub shift of finite type can be describe in terms of a matrix.

So, sub shift of finite type is transitive, if an only if this matrix a is irreducible and if we want to look into the proof of this theorem right the proof simply turns out by saying that when we are looking into transitivity what do we want to do we take a word u right we take another one w, we want to say that there has to be some word in between right because we know that our shift space is irreducible if and only if it is transitive it is a same thing. So, we want another word v coming up here in between u and v and what does it mean by saying that a is irreducible it means that whenever I take an i n j there exist an n such that this is strictly positive what does that mean right that basically means that you start with an what.

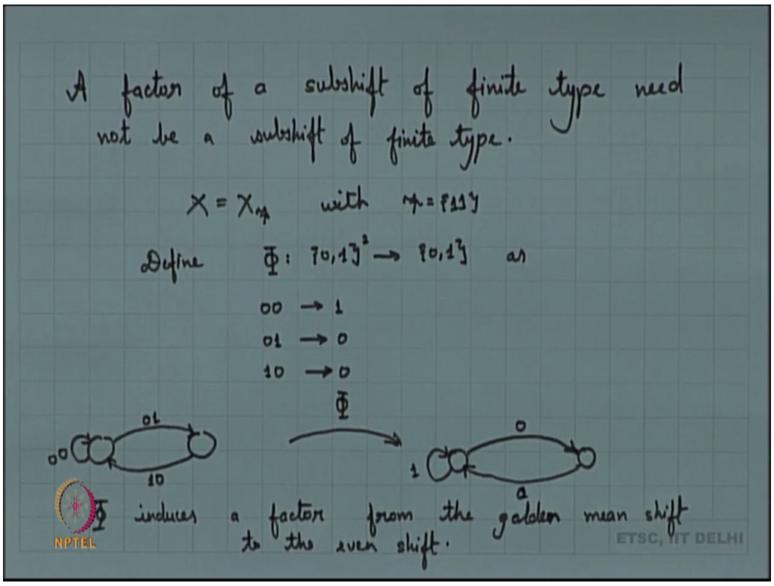
You start with the vertex i right, you take another vertex j there will always be a path between the vertex I to the vertex j right may be of length n right. So, this path will be n step. So, you will always get an n such that this is going to be positive and so, you will always get. So, given a word u given word v given a word u given a word w right you will always find of a word v of length n right such that this is a permissible word. So, basically saying that this is a sub shifts of finite type its same as saying that this is irreducible because my u could end up in the vertex right because this word u could perhaps end in the vertex I, your w is perhaps starting from the vertex j and in between you can always find a path; that means, you can find a word in between such that this word u v w right.

It is intermediating between the vertex I and the vertex j. So, these 2 quantities are equivalent and hence for us to characterize or to know whether a sub shift of finite type is transitive is enough it is enough to see whether the matrix happens to be irreducible we have a second risk theorem here also and I think the proof is almost similar. So, the second theorem here says that a sub shifts of finite type is mixing and when I call this mixing its basically strong mixing if and only if a is a periodic now this again has the same proof here because what you are doing is you are starting with u and w and you want to say that you can always sandwich a word of any length between u and v.

So, what happens here is we know that you start with any 2 vertices right you start with any 2 vertices your A N i j is always positive. Now once your is always positive right what you have here is that from vertex I right you start with a vertex I you go to a vertex j right and you can have any 2 vertices there will be a word of length n. So, if your u and z I w starts at j right you have a word of length n there, but then I could also think of u starting at some other vertex right and then again I have a length n plus 1 and. So, on and. So, what you get here is that this happens to be mixing if and only if it is a periodic there is another thing that we had seen and in the previous lecture.

We had seen that if I have a shift space which is conjugate to a sub shift of finite type then that shift space is also sub shifts to finite type because an M step of finite type if you take its conjugate image right that will give us some kind of n steps step sub shift of finite type depending on what is your conjugacy because your conjugacy is always coming through a block code , but maybe it is a very sad fact or maybe it is a interesting fact to see that a factor of a sub shift of finite type need not be a sub shifts of finite type. So, let us start to take an example here. So, all they want to say is that a factor of a sub shift need not be a sub shift of finite type.

(Refer Slide Time: 26:05)



Now, we can think of this example. So, let us try to take our simple golden mean shift standard example now let me define a block a block here. So, let us define. So, this is a 2 block code basically. So, I am defining that as 0 0 is mapped to 1 0 1 is mapped to 0 and 1 0 is mapped to 0, we have only 2 choices, right, it can be either 0 or it can be 1. Now let us look in to the higher block representation of the golden mean shift. So, we have seen that we can write it in terms of a edge shift here. So, we have this 0, we have 0 here right we have. So, we have 0 0 here we have 0 one here and we have one 0 here and then we have.

Now, this phi this is a 2 minus block thing this also gives us a graph homeomorphism right from this particular graph to this graph where we are preserving the vertices and all we are doing this we are taking the images of the edges. So, this gives us one this gives us 0 and this gives us again back one.

So, if you try to recall this factor right then if you try to sorry if you try to recall this part then recalling this part, we see that what happens here is that this also if I take by infinite work on this particular graph that is also going to generate a sub shift this is 0 yeah sure. So, this by infinite work on this is also going to generate a sub shift and this sub shift that will be generated since this graph is basically a factor of this graph right. So, this is also going to generate a sub shift the sub shift that is generated here will be a factor of this particular sub shift because this is this sub shift would be given by this particular block code.

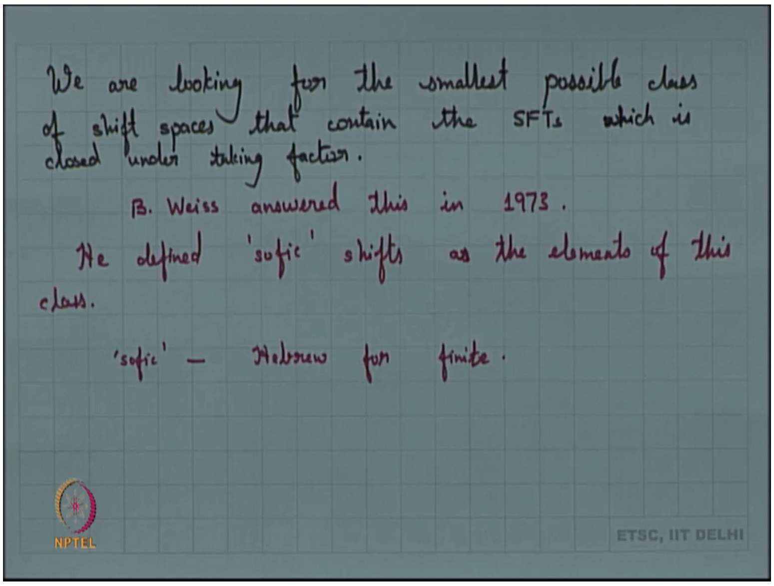
So, the factor this particular block code generates a factor right and this sub shift is basically obtained from this sub shift as a factor and that gives us that now if you try to see what is the sub shift turning out to be I have one I can have any number of ones possible, but once I take up a 0, I have you to again take back a 0. So, my 0 is always occur in e evens steps right. So, I will always have even number of 0s. So, this is basically my even shift you have even number of 0s between any 2 one. So, this is an even shift. So, what does this give me this? So, this phi generates a factor phi induces a factor from the golden mean shift to the even shift. So, the even shift can be realize as a factor of the golden mean shift.

Now, we know that the golden mean shift is a sub shift of finite type whereas; the even shift is not a sub shift of finite type. So, a factor of a sub shift of finite type need not be a sub shift of finite type now before we go up to something else I would just like to correct something that I did here . So, what we are having here is we have these vertices. In fact, when we are talking of edge shifts right we are not interested in what is what we label our vertexes, we are only interested in the labelling of the edges. So, what we do here is we try to take this as. So, we are calling this as a 2 we call this as a 2 a M plus 1 a M plus 2, right.

So, we are going into the higher block version. So, we call this as this factor and we call this as a 3 a 4 am plus 2 and a M plus 3. So, we are calling the edges, right. So, this is basically our higher block version of the edges. Now think of this factor. So, you have such a nice construction right such a nice kind of computational thing coming out; say a computational characterization of some kind of a dynamical system which is a sub shift of a finite type, but this sub shifts of finite type is not closed under taking factors. So, the question that comes up is can we have a smallest set of all shift spaces which also contain the sub shifts of finite type, but they are closed undertaking factors.

So, we are basically we have some kind of a class of sub shift of finite type and we know that this class is not closed undertaking factors. So, we want to enlarge this class of course, if we enlarge this thing, it becomes a full shifts. So, we want to enlarge this class in such a manner we want to add some more shifts there such that and that becomes closed under factors. So, this natural question was answered by very aptly answered by B Weiss.

(Refer Slide Time: 32:53)

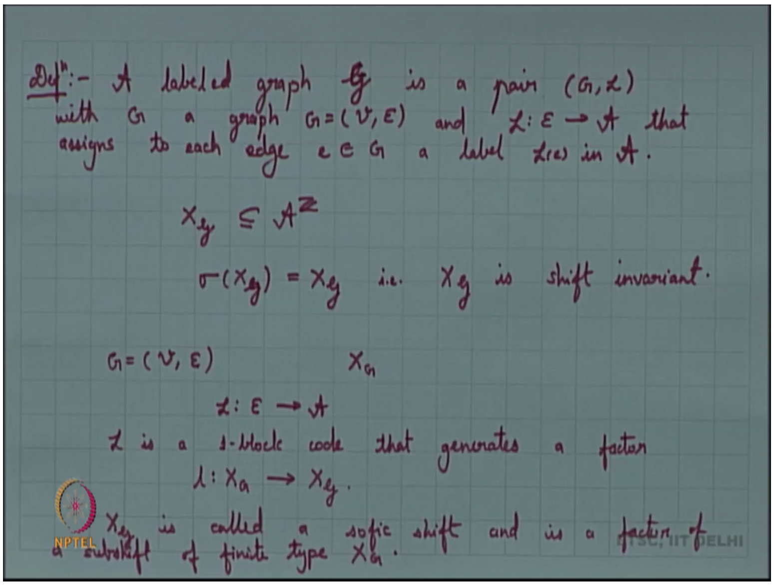


So, we look in to we will we are looking for a class we are looking for shift spaces that contain the sub shifts of finite type all sub shifts of finite type such that or which is closed under taking factors. So, if you take basically if you take any factor of an element in this class you remain in the same class. So, this was answered by B Weiss.

So, very aptly B Weiss answered this say in nineteen seventy 3 and he defined some special kind of shifts which he called us sofic shifts. So, he defined sofics shifts as the elements of this particular class now I just want say something about the word sofic here

So, if you look into the word sofic. So, this word ssofic is basically a Hebrew word right and this Hebrew word is for the word finite. So, there is some kind of a finite representation over here and that finiteness that feature of finiteness is captured in this part and that is why we called it as sofic shifts. So, let us try to look in to in the definition here. So, we start with a definition here. So, a label graph G.

(Refer Slide Time: 35:18)



So, let me call it as a labelled graph as this part this G script G a label graph G is a pair I have a graph here and I have a function here which I call it as labelling right with G to be a graph. So, G is my graph with v as a vertex set and e as an edge shift edge set and my l happens to be a function from this edge is a to the set a which we call it as an alphabet set. So, this this l is a function that assigns to each edge e in G right a label say we call it as l E right in a. So, a gives a label. Now what does that mean is now that we have a graph and in the graph right we have basically labelled each edge now look into the edge shift there.

If we try to look into the edge shift now right this edge shift is going to give us some kind of a labelled sequence. So, the edge shift gives us a labelled sequence where you can define this all your elements your sequences are defined in terms of whatever labels are there in a. So, if you look into this factor what we get here is this by infinite work on such an labelled graph is going to give what we call it as a shift right on this G and this basically will be a subset of a to the power z. So, you start with a finite labelling, supposing you say that you have you have a finite because your edges are finite your labelling anyway has to be finite here.

So, you start with your edges right for each of your edges you are basically giving some kind of a label and for that particular label you say that fine since there is a labelling here you think of the edge shift now. So, the edge shift now can be viewed as now a sequence of labels there right if you are taking a by infinite walk on the label graph. So, now, you have a sequence of labels right and you are generating a sequence of labels and that sequence of labels is what you call as a shift here shift coming up from the labelled shift and that is a subset of e to the power z.

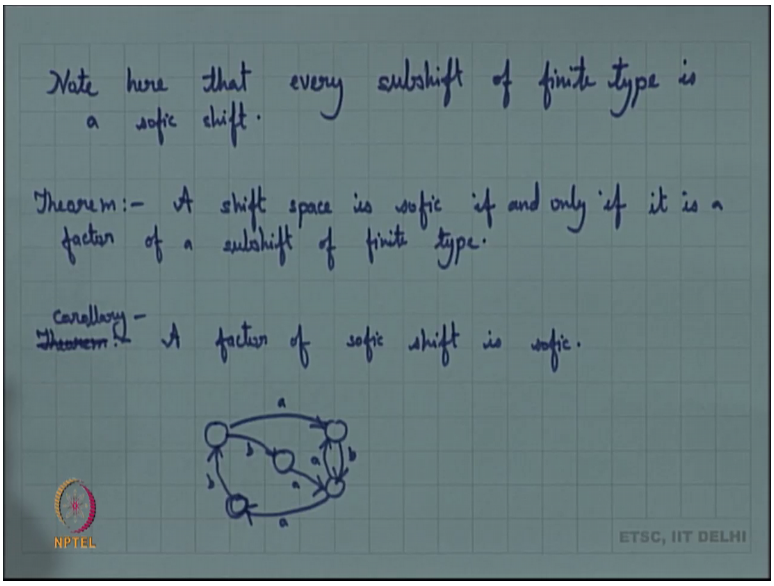
Now think of that if we look into this factor if I will take the shift of this labelled this what shift that we generate here right then this is shift invariant you see that your X G is shift invariant there is one thing right. So, this is shift invariant now thing of another factor here is that you already had this graph G right.

Which is having your vertex v and your edge is e and this graph G is going to generate an edge shift right which we call it as X G, the graph is this edge shift is already there with us now think of this edge shift and think of the space X G what we find here is that between e and a right there is some kind of a labelling. So, l is giving us a function between e and a now think of that we could think of l to be a block code. In fact, a 1 block code right each edge is being labelled by a. So, we can think of l to be a 1 block code. So, l is a 1 block code and what does it generate right, it generates a factor code right it generates a factor. So, let me call that factor to be l right it generates a factor from this X G to s g.

So, what do we have here we have X G substitute a finite type right and the new shift that we have here which is basically defined using the labelled graph happens to be a factor of this sub shift of finite type. So, what we have here is that this X G is called we call X G to be a sofic shift right. So, this X G is called a sofic shift and is a factor of the sub shift of finite type. So, what is this sub shift of finite type this is X G now think of that what we have here is that we have this sofic shift generated here by using the labelled graph now a labelled graph is just a graph with a labelling. So, your graph coupled with a labelling and that is going that is generating another shift space which we call as a sofic shift and it is very easy to see that.

This is shift invariant we know that all close sub shift invariant spaces or sub shifts right and since we can realize our sofic shift as a factor of the sub shift of finite type this sofic shift is closed. So, it is closed in invariant subset and hence it is a sub shift. So, what you get here is for by using this graph right you can generate something which is called a sofic shift now very interesting fact here is that when you have a sofic shift.

(Refer Slide Time: 42:24)



So, first of all we note here that every sub shift of finite type is a sofic shift do you all agree with me, can we say that every sub shifts of finite type is sofic we could have our labelling to be identity; is it not our labelling is identity and that gives us that every vertex.

Every sub shift of finite type we know that there is a way of writing every sub shift of finite type as a edge shift right and then we have this identity labelling, which makes it a sofic shift. So, every sub shifts of finite type is a sofic shift. The second factor is here we have seen something else some property of sofic shift, but let us look into this part a shift space is sofic if and only if it is a factor of a sub shift of finite type; we try look into the proof of this factor. So, we know that you start with a shift space right your shift space is sofic your shift space is sofic; that means, it is basically a shift space which is defined by a labelled graph right. So, you have a labelled graph which defines the sofic shift, but where is this level graph coming from.

Right its basically coming from that graph where you are labelling the edges so; that means, it is being generated by the one block code, which is your label; label map and that one block code is generating your what to say is generating a factor between the sub shift of finite type in the sofic shift. So, we know the shift space is sofic, if it is a factor of a sub shift of finite type. We need to look into the converse part, can we always say that this will be true think of that. If it is sofic, we know that it is a factor of a sub shift of finite type now think of a factor of sub shift of finite type what is a factor of a substitute a finite type? A factor of a sub shift of finite type right would be again I could think of this in terms of some kind of a factor map right. So, that is generated by some block code.

Now, if necessary we can run a higher block representation of the sub shift of finite type. So, that the block that we generate becomes one step, see all we need to define a labelling is a 1 block code. So, a if we have some factor of a sub shift of finite type then that factor of sub shift of finite type, right can be generated using some block code. Now if possible we go to the higher block version of a sub shift of finite type, then we can define our block code in such a manner that the block code is one step. Now once the block code is one step we can think of that block code as a label map, and that label map gives us a sofic shift right. So, a space is sofic shift then it has to be factor of sub shift of finite type, if there is some factors of sub shift of finite type.

It has to be sofic. So, we say that and since we know that every sub shift of finite type itself is sofic right basically we are getting into a bigger class which is sofic and again this is another theorem here. A factor of sofic shift is sofic do you all agree with that why should that be true. Just take the previous theorem this actually I should not write this as a theorem I should basically write this as a corollary, because you start with that if it something is sofic it is a factor of a substitute of finite type. So, if you take its factor then again that will be a factor of sub shift of finite type and hence it will be sofic, right. So, a factor of a sofic shift is always sofic.

So, now we have a very nice class of shifts which is closed undertaking factors right it is closed under and here and here again we can talk of some properties here, this sofic shift is a factor of sub shift of finite type we already know we can characterize the dynamical property of sub shift of finite type right by the help of just looking at the matrix there. So, if we have a irreducible matrix we get a sub shift a finite type which is transitive and hence we can say that if a sofic shift being a factor of a sub shift of finite type it is transitive whenever the sub shifts of finite type is transitive, it is mixing whenever the sub shift of finite type is mixing.

So, you have these characterization for the sofic shift, and I think we are not getting into details although one could start with this in various forms, but we are not getting into details here the only thing I want to note here is that if you try to take up say any sub any labelled graph right. So, your labelled graph would be something like that. So, you have edges here right. So, you have some vertices here and for every vertex there is right an edge right joining the vertices. So, I could have something going from here to here and it is possible to go from here back to here this is possible. Now if you look into this particular graph I always have a label here. So, I am labelling here a b it could be a here again b a right I have a here and I have b here. So, we have just some kind of a labelling of this graph. Now if you look into that factor this construction is similar to the construction of finite automata.

So, from the point of view of theory of computation right our sofic shift right nothing, but the language of sofic shift is just the language that can be generated from a finite automata and hence the language of a sofic shift is something which we can term regular. So, this is just some overlap with theory of computation here, but we are not getting going to get into details over here, and I think this is what we have for symbolic dynamics right we are not though there could be many things that we could do, but we will be using this as a tool as an example as a characterization sometimes, a hence forth in some lectures that we still have to do, but we are not getting into more deeper details over here. So, I hope this has been clear to all of you.