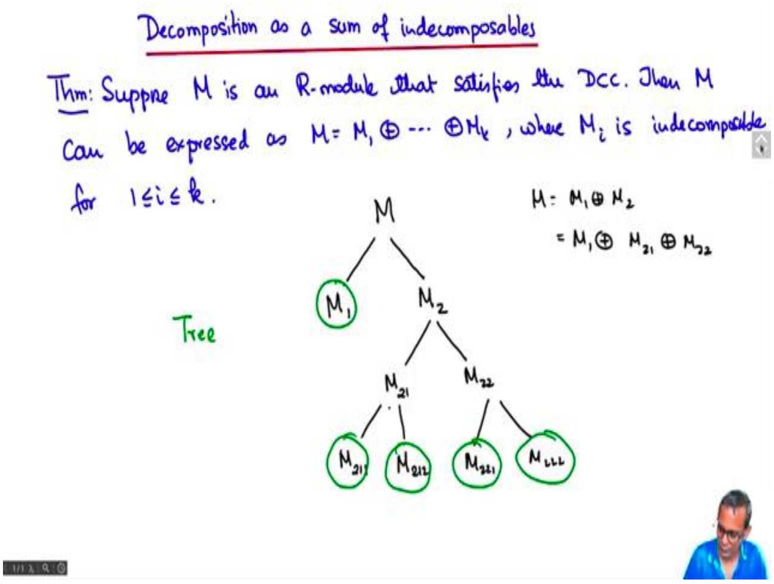
Algebra- II
Professor Amritanshu Prasad
Mathematics
Indian Institute of Mathematical Sciences
Lecture 81
Decomposition as a Sum of Indecomposables

(Refer Slide Time: 00:14)



In this lecture we will see how an module can be written as sum of indecomposables. So the main theorem is the following; suppose M is an R module that satisfies the descending chain condition. Then M is, M can be expressed as m1 plus Mk where each Mi is indecomposable for i between 1 and k. Now before I give you the proof, let me give you the proof idea. So the proof idea is the following; you start with M, and then well if it is indecomposable then you are done, if it is not indecomposable then you can break it up as a sum of two submodules both of which are non-trivial let us say M1 and M2.

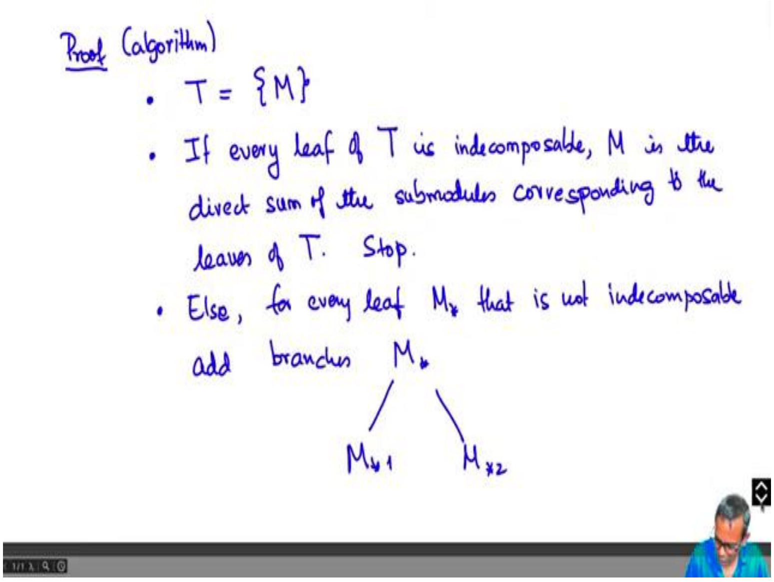
Now if M1 and M2 are both indecomposable then you are done, otherwise it is possible that otherwise one of them is not indecomposable. So let us say M2 is not indecomposable, so if M2 is not indecomposable then I can further decompose it into M21 and M22 and let us just for a moment assume that M1 is indecomposable so I would not do anything to it.

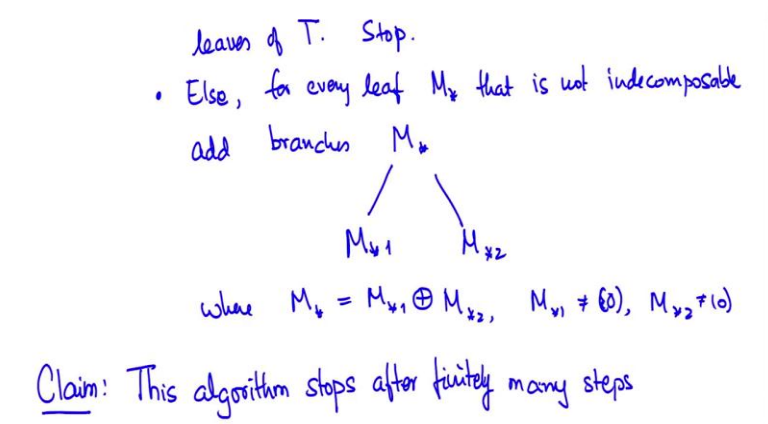
Now again if M1 M21 and M22 are indecomposable then what we have is, now note, we have M is M1 plus M2, but M2 is M21 plus M22. And well, you need to proof certain associativity property of direct sum decomposition which says that you can remove these brackets, but it turns out as M is now a direct sum of M1, M21, and M22.

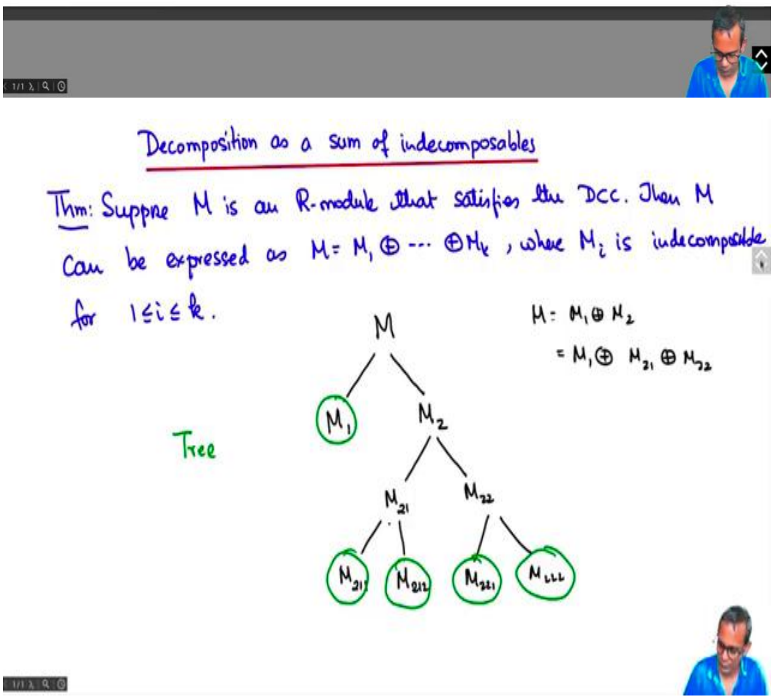
And now if these two were, M21 and M22 were indecomposable then we are done. We would have direct sum decomposition of M into indecomposable modules. But maybe they are not, so in that case what we would do is we would go M212, M211, M212, M221, M222. And then again continue this until you know the process must, until well either this process goes on in definitely it must stops after finitely many steps. We will see that the descending chain condition will ensure that this process stops after finitely many steps.

Now at each step there are certain modules, these ones, which are not branched yet. So either they are indecomposable or in the next step they will get branched. So these ones circled in green are called the leaves of the binary tree at that stage. So, this we can show that this process will stop then we will have an algorithm to write M as sum of indecomposables. So this whole thing is called a tree, it is called a binary tree and these nodes which are which do not have any branches, these are called leaves. so now let us just write down this algorithm a little more formally, so here is the algorithm.

(Refer Slide Time: 04:13)



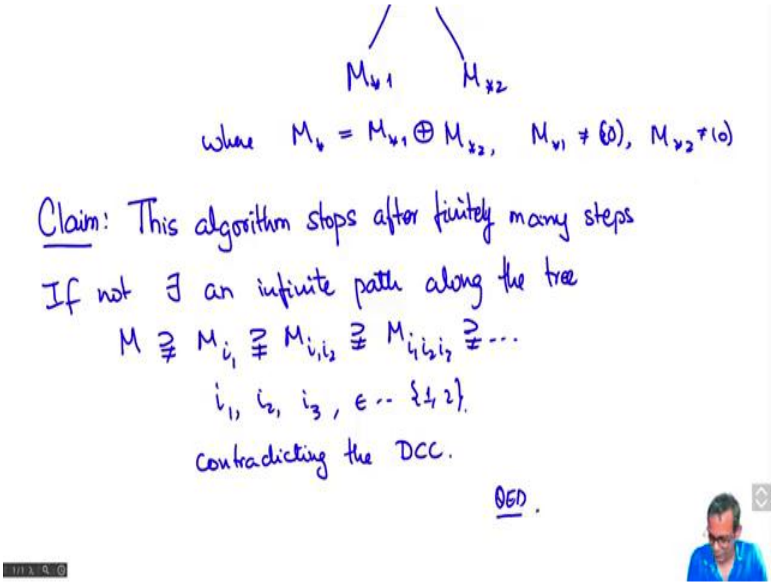




And we will show that, this algorithm stops by using the descending chain condition. So start with T equals just one node the tree with one node M, so at the first stage your tree has just this one node M which is a leaf. Now the algorithm goes as follows, if every leaf of T is indecomposable, T is the direct sum, M is the direct sum of the submodules corresponding to the leaves, and then you stop.

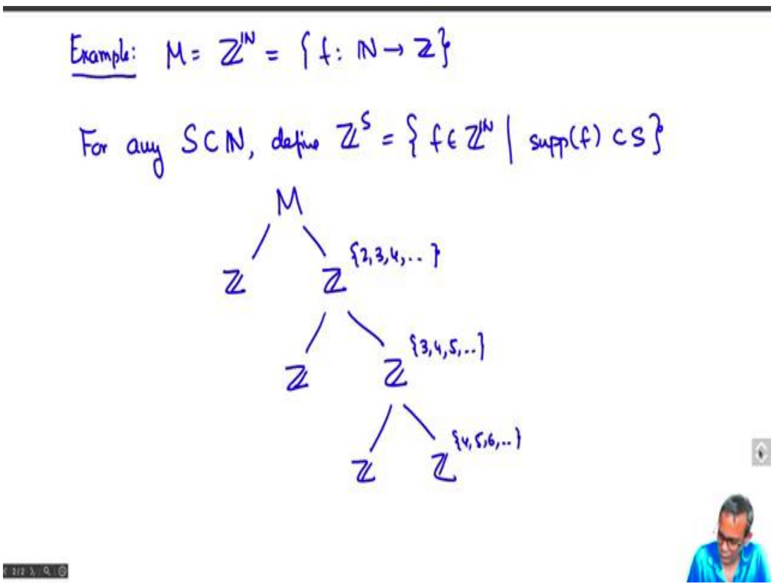
You stop with direct sum decomposition into indecomposables. Else, for every leaf M star that is not indecomposable, add branches M star below M star, which is namely M star 1, M star 2, where M star 1 and M star 2 give a direct sum decomposition of M star. So this algorithm just runs, and my claim is that this algorithm stops after finitely many steps.

(Refer Slide Time: 07:36)



Well, if not what will happen? If not there exist a path, an infinite path along the tree consisting of nodes M containing properly Mi once it would be either M1 or M2, which properly contains Mi1 i 2, which properly contains Mi1 i 2 i 3 and so on, where i 1, i 2, i 3 all belong to set 1, 2. And this would be strictly decreasing chain of submodules which would not stabilize contradicting the DCC. And so every module that satisfies the DCC can be written as a direct sum of indecomposables submodules. So this algorithm proves the claim.

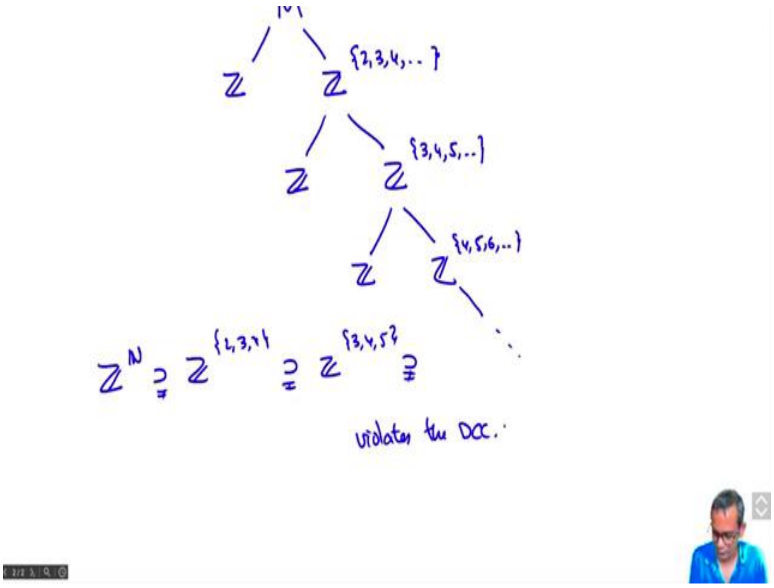
(Refer Slide Time: 08:59)



Let us look at an example. So let us take M to be Z to the power of n, which is nothing but the set of all functions from the natural numbers to integers and we think of it as a Z module addition is just a point wise addition of functions and multiplication of a function by integers is also just multiplication of its values. So, now for any S subset of N natural numbers define Z to the S be those functions F from natural numbers to Z such that support of F is continuous. That means F is 0 unless F of N is 0 unless N belongs to S.

Then for any S, so then what we can do is we can draw the following tree, so let us just run the algorithm, so M, well M is not indecomposable because I can write it as Z direct sum Z subscript 2, 3, 4. This is the value of the, so, this decomposition I just think functions which supported only at 1 and so these are the functions supported at 2, 3, 4, and so on. And then this can be written as Z direct sum Z 3, 4, 5, and so on. And then this again can be decomposed Z 4, 5, 6.

(Refer Slide Time: 10:47)



So what this example illustrates is that when this algorithm runs forever without stopping then here you see along this branch of the tree we get descending chain of submodules that never stabilizes. So the chain Z to the n contain Z to the 2, 3, 4 contains Z to the 3, 4, 5, this violates the DCC.