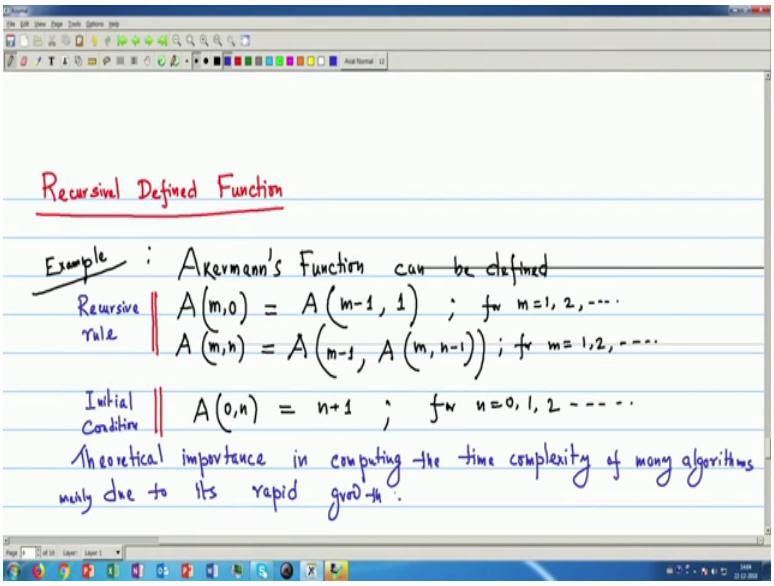
Discrete Structures

Prof. Dipanwita Roychoudhury
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 28
Recursion [Contd.)

So, in the introduction of Recursion we have seen or we have learned that how actually recursively defined functions are constructed. Now, today we will see that thing with an example that how the recursively defined functions are constructed and their properties are used.

(Refer Slide Time: 00:49)

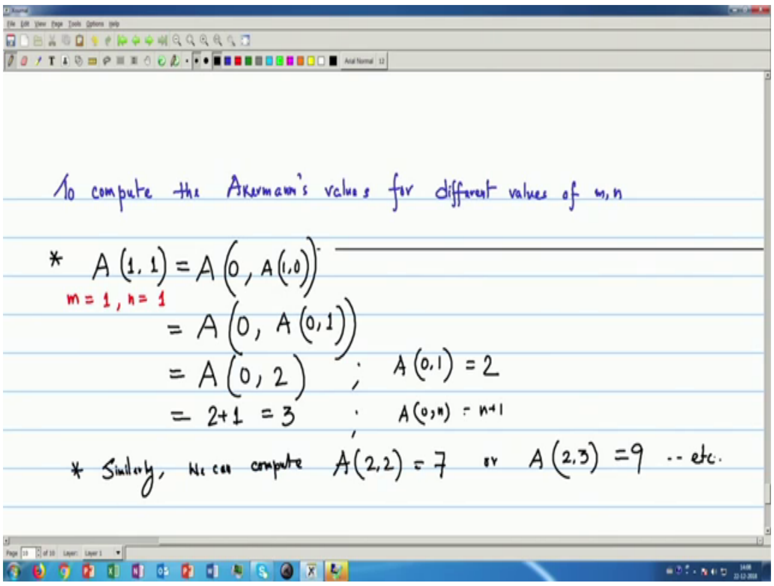


Now, in the last lecture we have seen the recursively defined sequences. Now we will see the how it is though we define that functions recursively ok. Now, we take one example then examples illustrate this thing. Now, we take the example of Ackermann's function and this can be defined this is some simplified form of the general Ackermann's function and it is defined as the say A m n all first we give it is defined in two steps. So, first A m 0 is a m minus 1 1 for m equal to 1 2 for all integer values.

And the second step is A m n is A m minus 1 A m n minus 1, for m equal to 1 2 like that. See this is together these two steps are actually my recursive rule this is my recursive rule and the initial values the if I take the initial condition initial condition this is A 0 n equal to n plus 1 and this is for n equal to 0 1 2 like that. So, this is my so together these initial condition and the recursive rule the Ackermann's function is defined.

And say it is of theoretical importance because that it is the rapid growth of the values of the functional values of this Ackermann function. And it appears in many algorithms when we compute the time complexity it appears. That is why it is a importance. So, the theoretical importance is so it has theoretical importance it in computing; the time complexity of many algorithms. And this is mainly due to it is mainly due to it is rapid growth and this is mainly utilized to do that thing. Now, we see that how the for some fixed values of m and n how it is evaluated using or how it is computed using the Ackermann's function say.

(Refer Slide Time: 06:27)

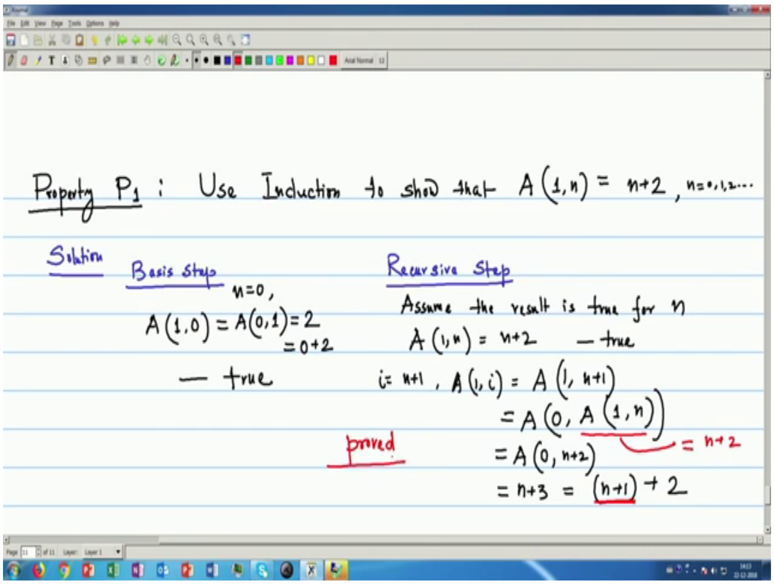


So, we can write to compute we can call this is Ackermann's value for different values of m and n. See we one we see that how to compute A 1 1 this is A 1 1. So, if we apply the recursive rule we get A m minus 1. So, here my m equal to 1 n equal to 1. So, if I apply m minus 1 this becomes m minus 1 this becomes 0 and A A m minus 1 A.

So, if you see that thing m minus 1 A n m minus 1. So, we write in the next step at A 0 A 1 0. Now, that A 0 and if I apply in this way this becomes A 0 then A 0 1. So, now if I apply the second step the initial condition that A 0 n then we get A 0 1 A 0 1 equal to 2 since A 0 1 equal to 2. Now A 0 n is n plus 1 so, this becomes 2 plus 1 equal to 3 since A 0 n is n plus 1.

So, in this similar way we can get that my A 2 2 similarly we can get is A 7 or A 2 3 is 9 etcetera. Now, the similar way as we have done for recursively defined sequences. Here also we can give some property for recursively defined functions and we can use we can prove using mathematical induction.

(Refer Slide Time: 10:21)



So, one such property we see; we give you a property its P 1 use induction to show that A 1 n equal to n plus 2 for n equal to 0 1 2 like that. Now, since it is induction so we can give the solution by taking the basis step and the recursive step. So, we see the basis step say for n equal to 0 this is A 1 0. This becomes A 1 0 and A 1 0 is if I apply the rule then this becomes A 0 1 which is 2 and this equal to 0 plus 2; that means, the (Refer Time: 12:15) it is true. So, basis step is proved.

Now, if we see the recursive step we see now the recursive step. Now, we assume that it is true for n assume that it is true for the result is true for n; that means, A 1 in equal to n plus 2 is true. Now, we have to show that for n plus 1i equal to n plus 1 also it is true. So, for I equal to n plus 1 I can write A 1 i is A 1 n plus 1 that if I apply the recursive definition of Ackermann’s formula. So, this becomes A 0 A m n minus 1 so 1 n.

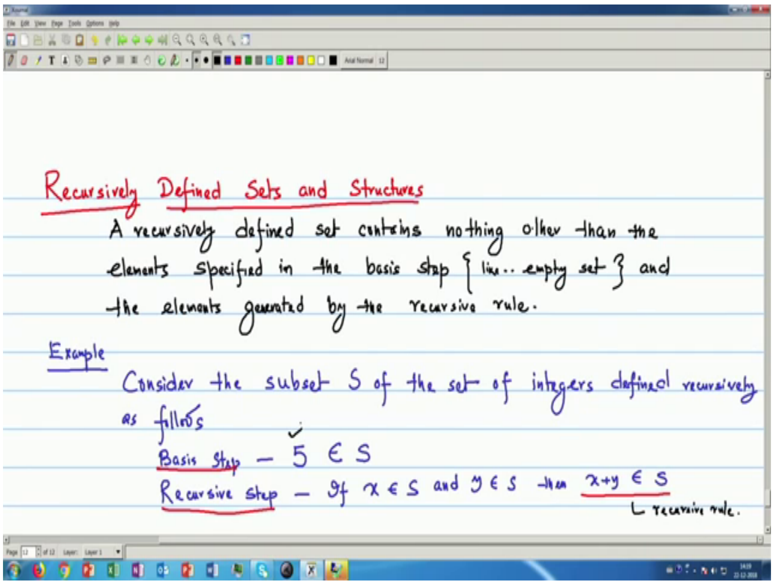
Now, according to the principle of mathematical induction that A 1 n, this conjecture is already true and that value equal to n plus 2. So, this becomes A 0 this becomes A 0 n plus 2. When A 0 n is n plus 1 so this becomes n plus 3; that means this is equal to n plus

1 plus 2. So that means when I equal to n plus 1 when I equal to n plus 1 this is n plus 1 plus 2. So, this is properties proved.

So, that what we see the totally similar way that the recursively defined functions the recursively defined sequences that we can frame the recursive formula. And then with the formula can also be proved by using the mathematical induction and the recursive definition of the sequences or the function.

Now, we can see that how we can define the recursive or we can define recursively the sets and structures. In the introductory lecture of recursion we have seen with some examples, but now we see that how it is actually framed.

(Refer Slide Time: 15:43)



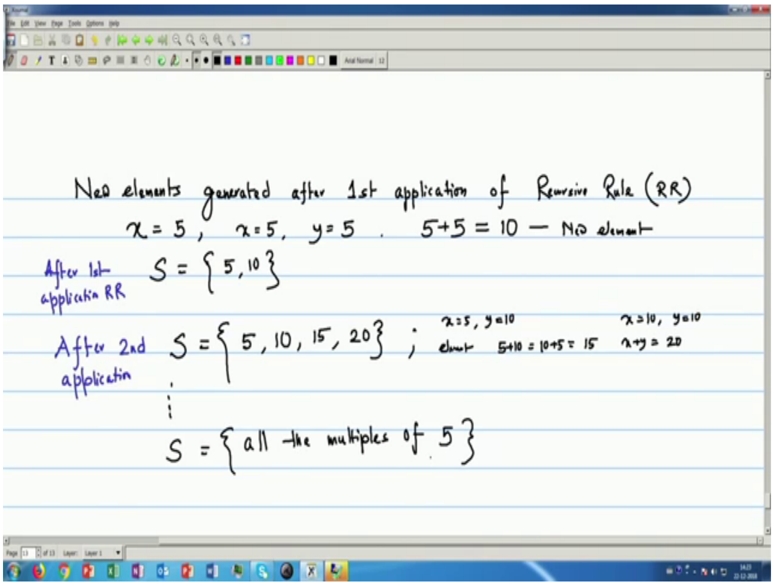
So, recursively defined sets and structures. So, recursively defined sets that we can start thus with the empty set. And then we will define a specific rule and we will generate the new elements so this is the main concept.

So, the definitions specifies or we can write the that it recursively defined set other than the elements specified in the basis step. It can be empty set like it can be like it is it can be empty set or some trivial values and the elements generated by the recursive rule. We take one very simple example and we illustrate that how the sets are constructed recursively.

So, consider a subset is of the set of integers defined recursively as follows. So, I can write that since now we know that it has it must have two step. So, basis step is say the instead of empty set I take one element 5 and these 5 belongs to S and what is my recursive step recursive step that if x belongs to S and y belongs to S then x plus y belongs to S. So that means all new elements will be generated by this formula. So, this is my basis step this is my recursive step and this is my recursive rule. So, my recursive rule is this.

So, as already I mentioned my set S consist only the initial element or elements defined by basis step; that means 5 and the newly generated elements by the recursive rule. So, what the set becomes.

(Refer Slide Time: 21:21)

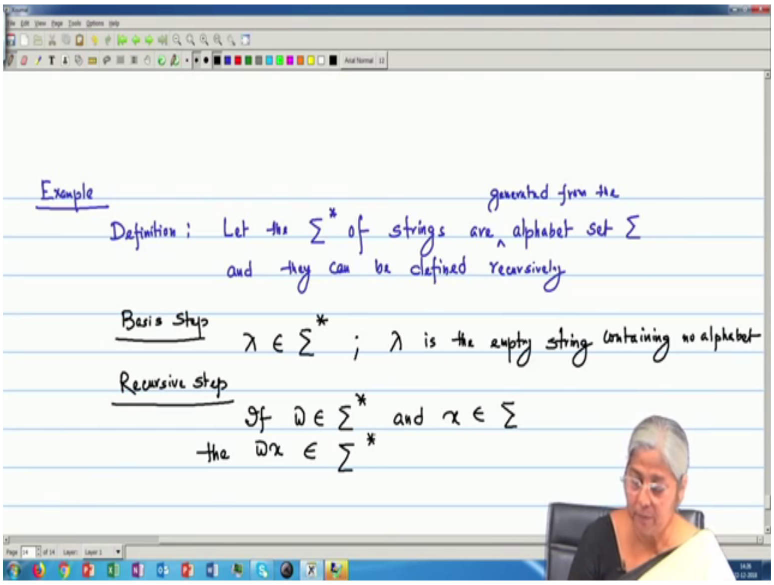


So, if I tell that new elements that generated elements generated after 1st application of recursive rule. Then though x is here only 5 I have only one element so I can think x equal to 5 y is also 5. So, the newly generated element is 5 plus 5 is 10 so this is the new element.

So, after the first application this becomes 5 and 10 ok. Then after the second application so after first application of recursive rule we can write R R as if this is recursive rule, then after 2nd application I can write my S equal to 5 10 15 25 because now x equal to 5 y equal to 10. So, the new element is element is 5 plus 10 equal to 10 plus 5 equal to 15.

Now, x can be 10 y can be 10 so x plus y becomes 20. So, if I now proceed in this way the finally, what will get that S is the multiple of S is multiple all the multiples of 5. I get S is a set for all the multiples of 5. So, one set we can generate by using this type of recursive definition.

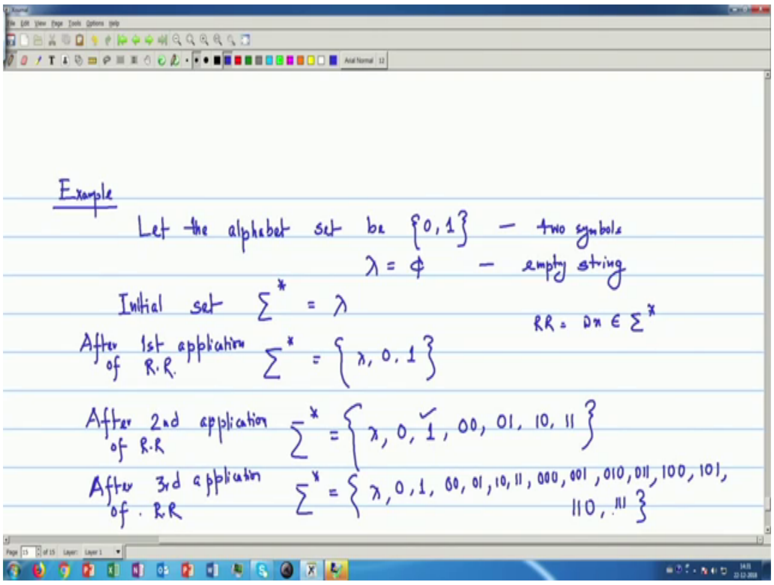
(Refer Slide Time: 24:49)



Now, we take another example that how it is used for generating string. So, first I give a definition I give the definition of string. Let the sigma star of strings are alphabet set sigma. And they can be defined that generated from the alphabet set. And they can recursively or they can be defined recursively as follows. See the basis step is and define the basis step say lambda belongs to sigma star.

Defined lambda is the empty string lambda is the continue no alphabet containing no alphabet ok. What will be my recursive step? See if w belongs to sigma star and x is an alphabet; that means, x belongs to the my alphabet set then new string can be generated as w x belongs to sigma star ok. Now, if we with this definition if we take the example.

(Refer Slide Time: 28:13)

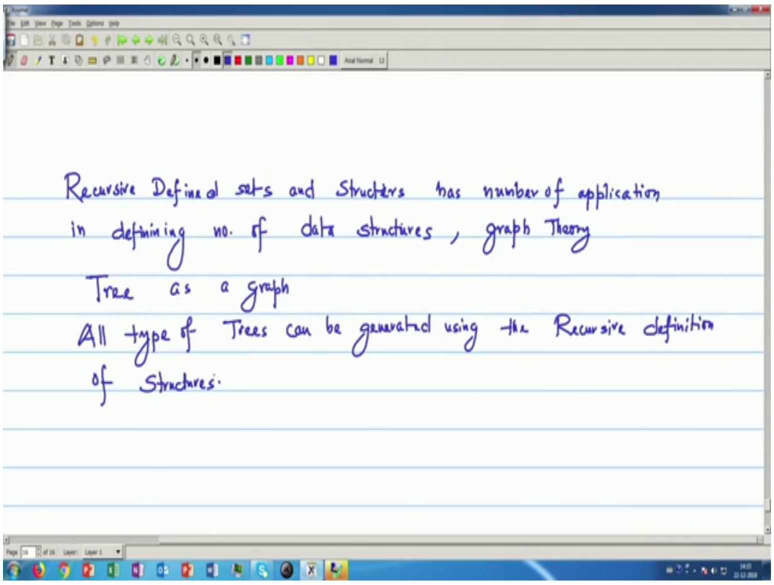


Where the example with any alphabet set the fixed alphabet set. So, let the alphabet set be 0 1 I have only two symbols ok. And my lambda is the empty set is the empty set, I write empty string containing no symbol how we generate that what will be the set.

So, the initial set is I can write sigma star is lambda. Then first application after first application of the recursive rule that sigma star becomes the set lambda 0 and 1 ok. Then after second application of the recursive rule I get sigma star is the empty set lambda 0 1 then w x we if we remember the recursive rule the w x belongs to sigma star. So, here x is 0 1 so, I get 0 0 0 1 1 0 11.

Now, after 3rd step 3rd application again it will be concatenated. So, I will get with strings of length up to 3; that means, 0 1 0 0 0 0 1 1 0 1 1. And now I can with all these I can append z 0 so, this is 0 0 0 0 0 1 0 1 0 0 1 1. Same thing I can now do for these one symbol so, this becomes 1 0 0 1 0 1. Then 1 1 0 1 1 1 and in this way I can continue. So, this is the way how actually we can do that thing. Now, this is how sets are defined in are defined recursively and with new sets are constructed.

(Refer Slide Time: 32:39)



So, it has a application in computer science and in mathematics it is these recursively defined application sets and structures. I have number of applications this number of applications in defining different data structures in defining number of data structures. And it is the importance same graph theory say if we can define tree as a graph, then this trees all different type of trees that can be generated.

All type of trees can be generated using this recursive definition of structures. So, what we have read that particularly the set the functions the sequences for that if we have some initial values. And we can give a recursive rule and then by applying the recursive rules that we can construct or we can generate the new sets as well as that new functional value of the terms of the sequences using this recursion very easily. And it has a number of applications particularly in combinatorics in graph theory in defining data structures.

And with this we finish the how recursively defined sets functions subsequences can be constructed along with the concept of principle of mathematical induction.