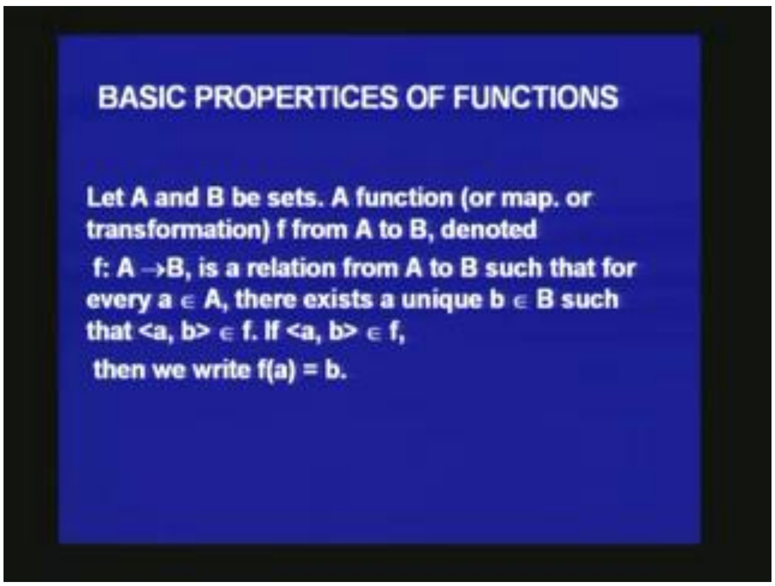
Discrete Mathematical Structures
Dr. Kamala Krithivasan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Module-1
Lecture # 24
Functions

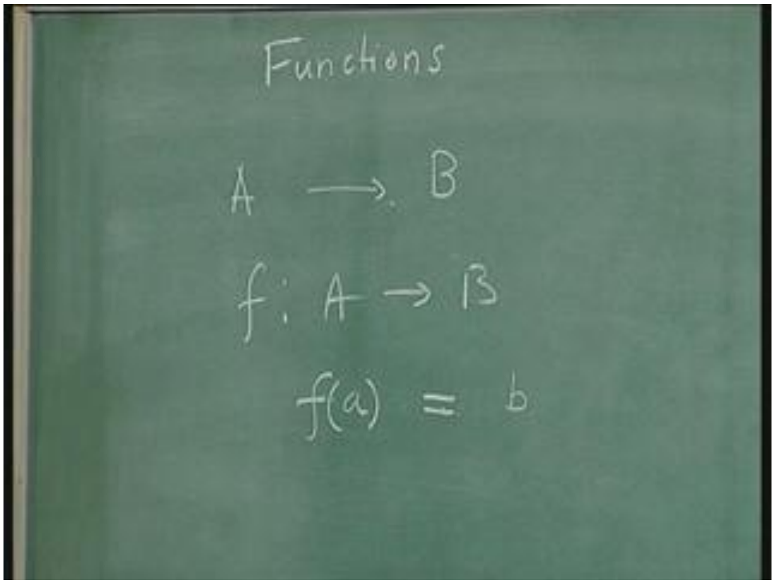
Today we shall learn about functions. A function from a set A to a set B is a rule which connects an element of B with every element of A. So it is a rule which specifies one element of B with each element of A. Let us see the formal definition of function.

(Refer Slide Time: 01.40)

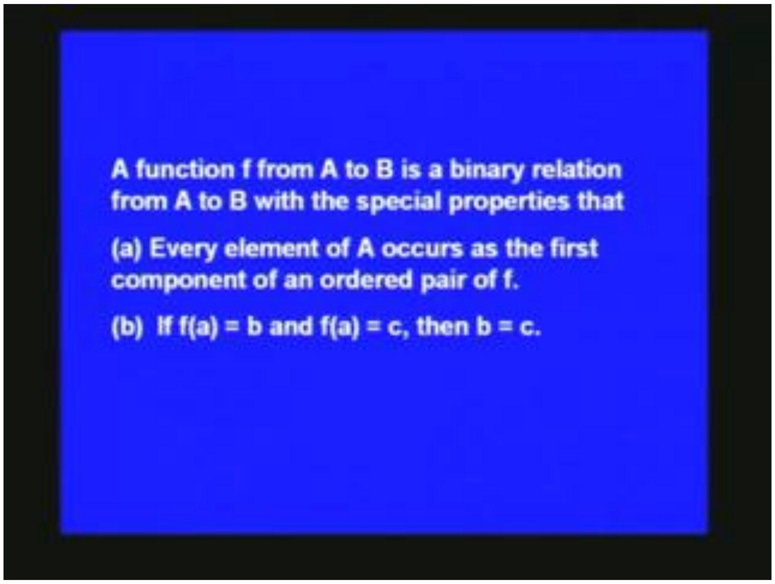


Let A and B be sets. A function or map or transformation f from A to B denoted f: A arrow B is a relation from A to B such that for every a belonging to A, there exists a unique b belonging to B such that a, b belongs to f. if a, b belongs to f then we write f a is equal to b. So it is a map or a transformation which is usually denoted like this. And with every element of a you associate a unique element of b and this you denote as f a is equal to b and it is a particular case of a relation. Relation is ordered pair so the first component will belong to A and second component will belong to B. Here there are some restrictions on this relation which defines a function.

(Refer Slide Time: 02.37)



(Refer Slide Time: 02.51)



A function f from A to B is a binary relation from A to B with the special properties that every element of A occurs as the first component of an ordered pair f and if f a is equal to b and f a is equal to c then b is equal to c. These two properties must be satisfied by the definition of a function, it is a particular case of a relation. Now let us see some examples of what are function and are what are not functions.

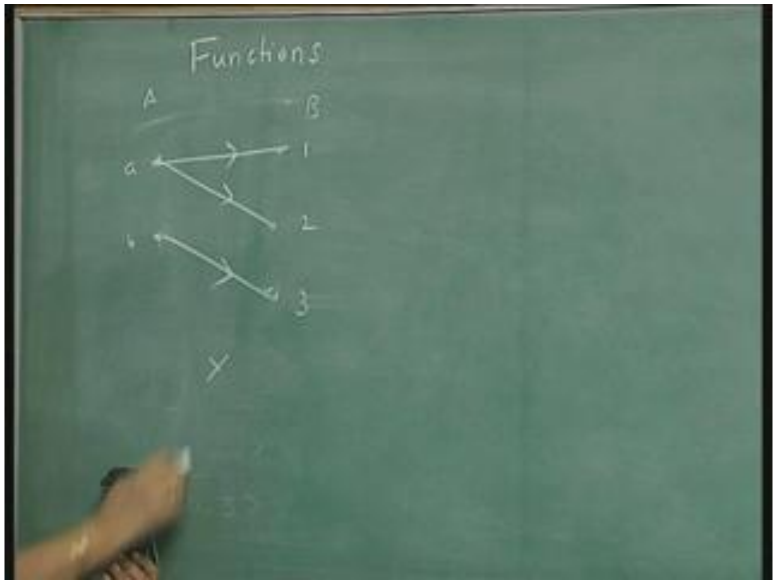
Consider two sets A and B and A has say elements a, b and B has elements 1 2 3. Then if you say like this; this is an example of a function. And another example would be on the same sets.

Here you can say f(a) is equal to 2 and f(b) is equal to 3. So these are examples of functions. If you represent it as a relation it will be represented as ordered pairs so you will have a2 b3 as ordered pairs. Every element of a will be the first component of some ordered pair and you can have only one element with a as the first component you cannot have something else. So let us see some more examples. If you have this like this this is also an example of a function, this is also a function, here f(a) is equal to 2 fb is also is equal to 2.

(Refer Slide Time: 04.54)

What are not examples of functions?
Look at this, the same set where A consists of two elements a and b and B consists of three elements 1 2 3 and A consists of two elements. Now if I define like this this is not an example of a function because for b we do not have a map at all. Only a occurs as a first component of an ordered pair for b we do not have anything so this is not a function.
Now you look at this, there are two ordered pairs a1 and a2 a occurs in two ordered pairs as the first component. Then also it is not a function, a is mapped on to 1 and also that is not possible, a should be mapped on to a unique element so this is not a function.

(Refer Slide Time: 05.58)

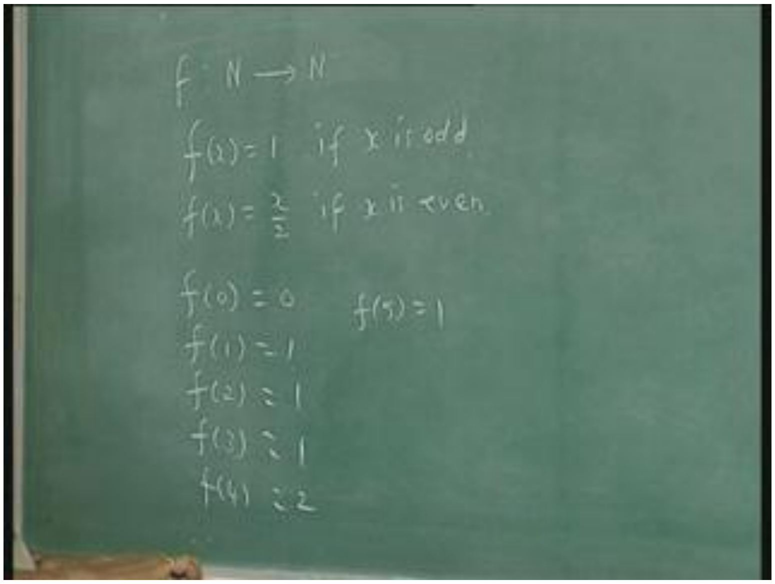


Sometimes this is called like this; this is called a partial function we will come to that a little bit later. So we have two conditions; every element of A occurs as the first component of an ordered pair f and if f a is equal to b and f a is equal to c you cannot have two different elements b and c so b should be is equal to c. Actually, you have come across functions in schools, you would have studied about several functions, some continuous functions, differential build functions and so on from the set of real numbers to the set of real numbers they are all examples of functions only.

But here, after learning relations we are looking into functions as a particular case of relation and we will also be looking at some special properties and what are inverses of functions and things like that.

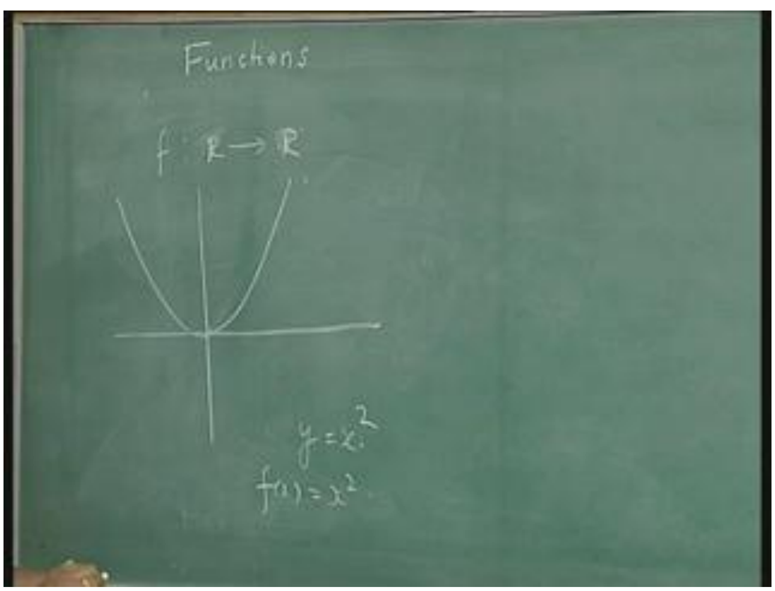
For example, take the set of natural numbers to natural numbers. A function from non negative integers to non negative integers, you can define like this f(x) is equal to 1 if x is odd you can define this way and f of x is equal to x by 2 if x is even. In that case what can you say about f(0)? F(0) is even so 0 by 2 is 0, f of 1 is 1 because it is odd, f(2) is even so it is x by 2 1, f(3) is odd so it is also 1, f(4) will be 4 by 2 that is 2 and f(5) will be 1 because it is odd and so on, this is one example.

(Refer Slide Time: 08.06)



Any graph if you take that also represents a function like you know if you take the set of real numbers this is a real number this is a real plane and you draw a line like this, this represents the line y is equal to x or this represents the function f(x) is equal to x. And if you take some function like this some graph this represents the y is equal to x square and this represents the function f(x) is equal to x square from the set of real numbers to the set of real numbers. Like that you can represent real functions from real to real as a graph in the two dimensional plane.

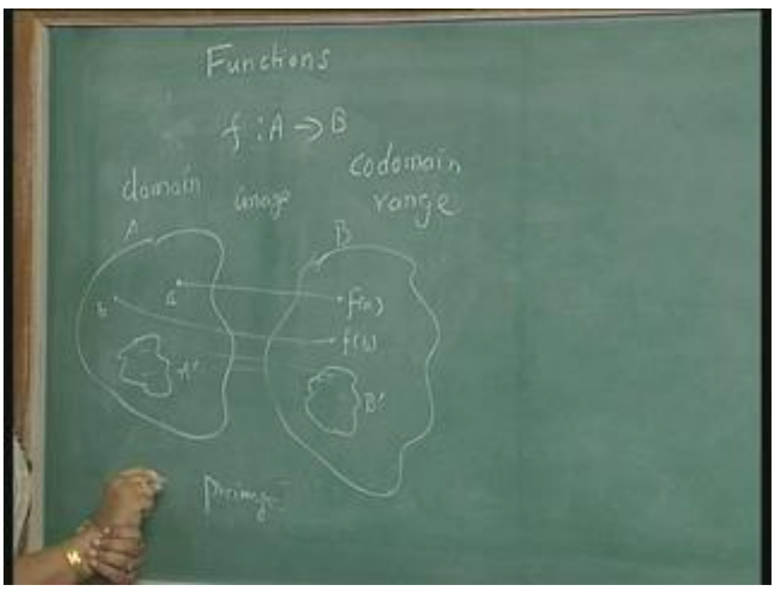
(Refer Slide Time: 08.56)



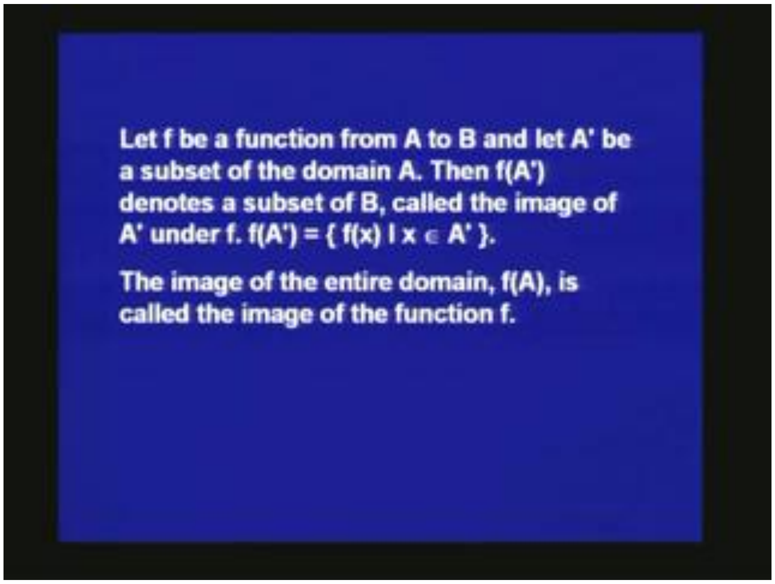
Now you have a domain A and a codomain B. in relation when you define a relation on A cross B you call A as the domain. Here also, for the function also you use the same terminology you call A as the domain and B as the codomain and sometimes it is also called as range, B is called the codomain or the range. So, if you have the set A and another set B what does the function f do? f is from A to B where A is the domain and B is the codomain. Now, take an element a this will be mapped onto an element f(a). Take another element b this will be mapped onto an element f(b) here.

F(a) is the image of a and f b is the image of b. You use the word image, f(a) is the image of a, f(b) is the image of b, a is the preimage of f(a), you can also use the word preimage, so a is the preimage of f(a) and so on. Now, if I take a subset here denoted by A dash each element of this will be mapped here and all these images if you take that will be a set B dash. So you can also look at this function as mapping subsets of A into subsets of B. Function maps a into individual, each element of a into individual elements of b. But you can also look at it this way, you can look at it as mapping subsets of a into subsets of b, this is what we will consider now.

(Refer Slide Time: 11.05)



(Refer Slide Time: 11.17)

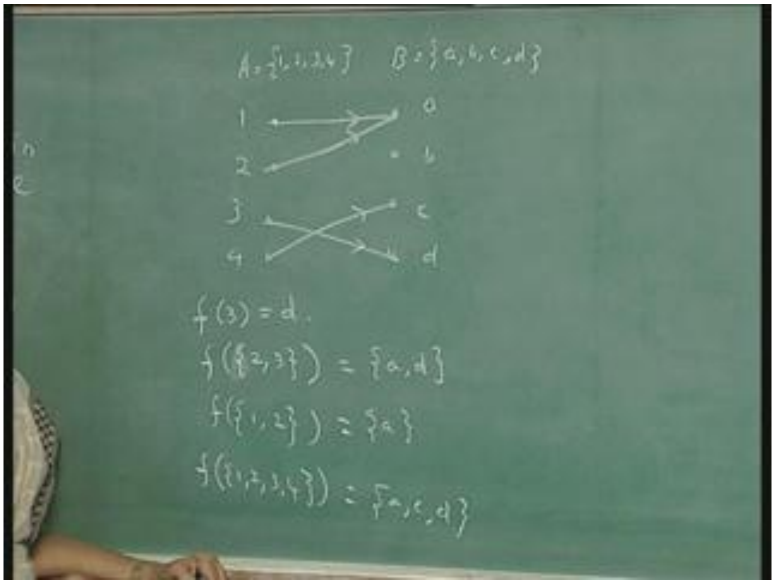


Let f be a function from A to B and let A dash be a subset of the domain A. Then f(A dash) denotes a subset of B called the image of A dash under f. That is f(A dash) is equal to f(x) where x belongs to A dash. The image of the entire domain f(A) is called the image of the function f. If you have a function f mapping A to B then you have a subset of A dash of A then this is mapped onto a set B dash this A dash is mapped onto B dash and B dash is called the image of A dash under f. And the whole set, I may have this B and A. If you consider all the elements mapping on they may be mapped on to the whole set B or sometimes it may be a subset of B then this is called the image of A under f.

For example, if you take these three elements and three elements here a b c and 1 2 3 a will be mapped on to 1, suppose b is also mapped on to 1, c is mapped on to 2 then the image of the whole set is only 1 2 a subset of this set. Now let us consider some example and explain this concept. Suppose you have a set A is equal to 1, 2, 3, 4 and B a set is equal to a, b, c, d so you have something like this 1 2 3 4 and a b c d, a map is defined like this 2a, 3d, 4c like that, this is a function.

Now what can you say about f?
If you take individual element you can say like this; f3 is d and so on. If you take a subset f of what is f of 2 3? Here 2 is mapped on to a and 3 is mapped on to d so this will be a, d. So you can look at it as mapping subsets into subsets. If you take f of 1, 2 where 1 and 2 are both mapped on to a so you will get a. if you take the whole set 1, 2, 3, 4 then 1 and 2 are mapped on to a, 3 is mapped on to d, 4 is mapped on to c you get a, c, d this is the image of this, this is the image of this and so on.

(Refer Slide Time: 14.56)

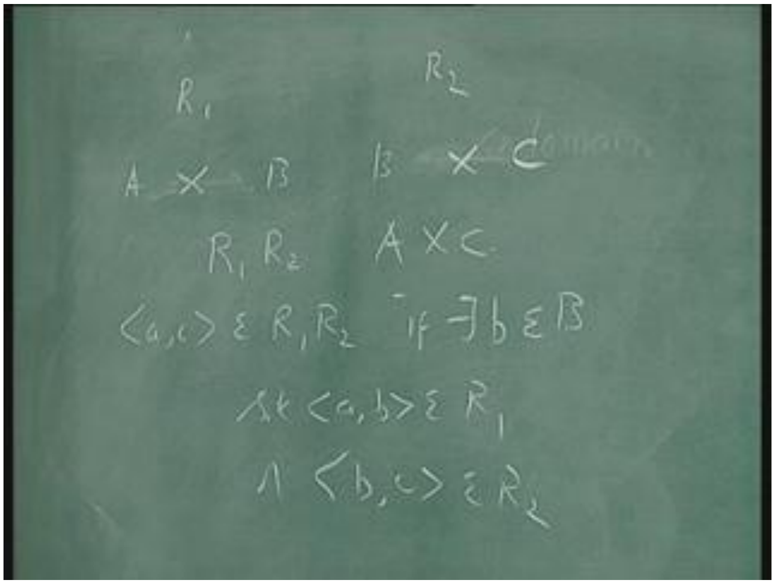


Next, we shall consider composition of functions. You know that you can combine relations. You have composition of relations R_1 and R_2, you have R_1 and R_2 and you talk about the compositions.

How do you combine a function?

That is the next point we will consider. Suppose you have a relation R_1 on A cross B and R_2 on B cross C then R_1 R_2 specifies a relation on A cross C. And how is that defined? a, c belongs to R_1 R_2 if there exists b belonging to B such that a, b belongs to R_1 and b, c belongs to R_2 this is how we define composition of relation. Now how do you define composition of functions?

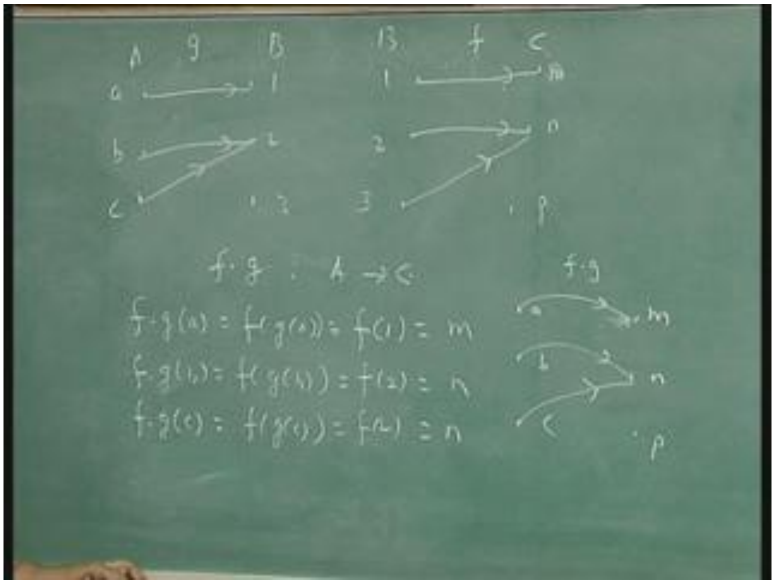
(Refer Slide Time: 16.07)



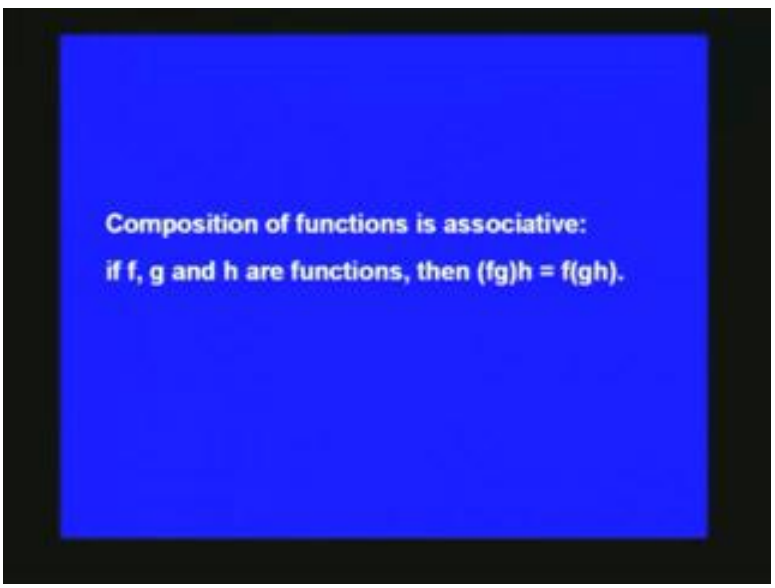
Now, let g be a function from A to B, g is a function from A to B and f is a function from B to C. Then f dot g you can use this symbol or just sometimes just write fg, this is a composition of functions it is defined like this f. g it is from A to C it will be a map from A to C and it is defined like this; f dot g(x) is equal to f(g(c)). So if you have a function, now g is from a b c to 1 2 3 something like this and f is from 1 2 3 to some x or m n p, then 1 is mapped on to m, 2 is mapped on to n, 3 is mapped on to n then what is f dot g? It is from A to C, A to B this is B to C so it is from A to C.

You have to define each element and it is defined like this. So f. g(a) is equal to f(g(a)) and what is that f of what is g(a)? It is f(1) and what is f(1)? That is m. So f(g(a)) is m, f. g(b) will be f(g(b)) and that is f of what is g(b)? It is 2 so 2 is mapped on to n. What is f. c? That is f(g(c)) that is f of what is g(c)? That is 2 so that is n. So this composite relation will be represented by a diagram like this a b c m n p, f(a) is this is f. g and f(b) is n, f(c) is also n it will be mapped on like this, this is known as composition of functions.

(Refer Slide Time: 19.34)



(Refer Slide Time: 19.39)

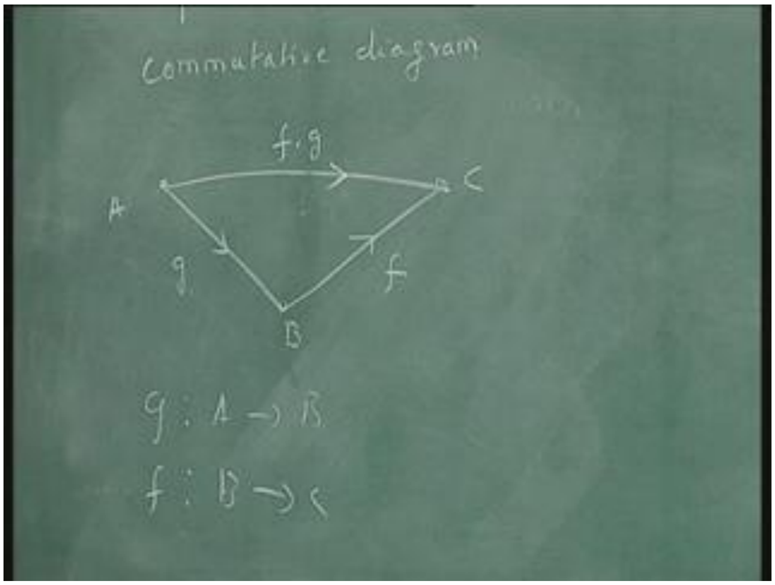


Now this is represented by a diagram known as the commutative diagram. Composition of functions is denoted. We have considered composition of functions to represent that we use a diagram known as the commutative diagram, how is this represented?

For example, A to B you have g, g is a function which maps on to A on to B. Then B to C you have a function f. Then when we combine f. g denotes a function from A to C. So, this is denoted as f dot g then f dot g will be from A to C. Please note that it is not gf it is fg. So from A to B you have a function and from B to C you have a function f then f dot g is marked from A to C.

Now, composition of function is an associative property. Here you must note that if you want to have this composition g is from A to B and f is from B to C. If these are different then you cannot talk about the compositions. In the composition of relations also we have seen that when you want R_1 R_2 to be combined then the co domain of R_1 should be the same as the domain of R_2. Similarly, if you want to have the composition of functions the domain of f should be the same as the codomain of g.

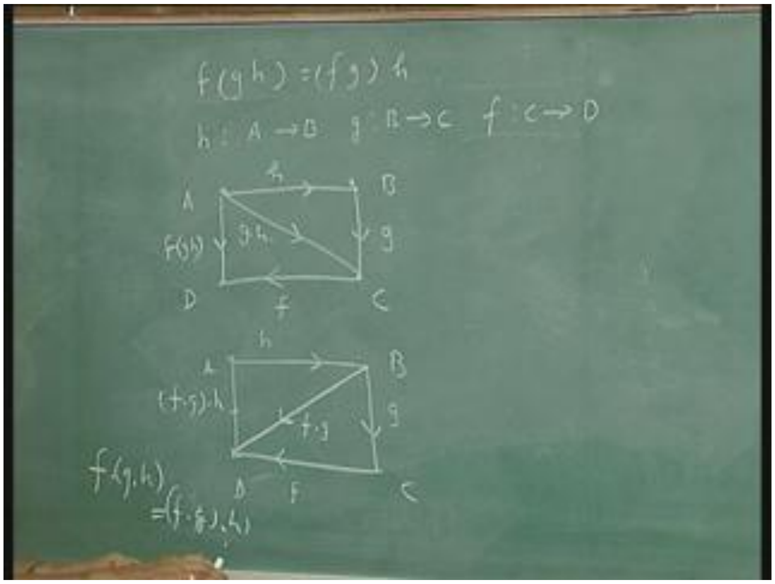
(Refer Slide Time: 21.23)



The composition of function is associative. Suppose f, that is you want to show that f(gh) is equal to f(g(h)). Now, h is a function from A to B, g is a function from B to C and f is a function from C to D. In the commutative diagram you will represent like this; from A to B you have h from B to C you have g and from C to D you have f. Now, how do you denote this? This denotes g dot h and this denotes g dot h and this denotes f so this will be denoted by f(gh).

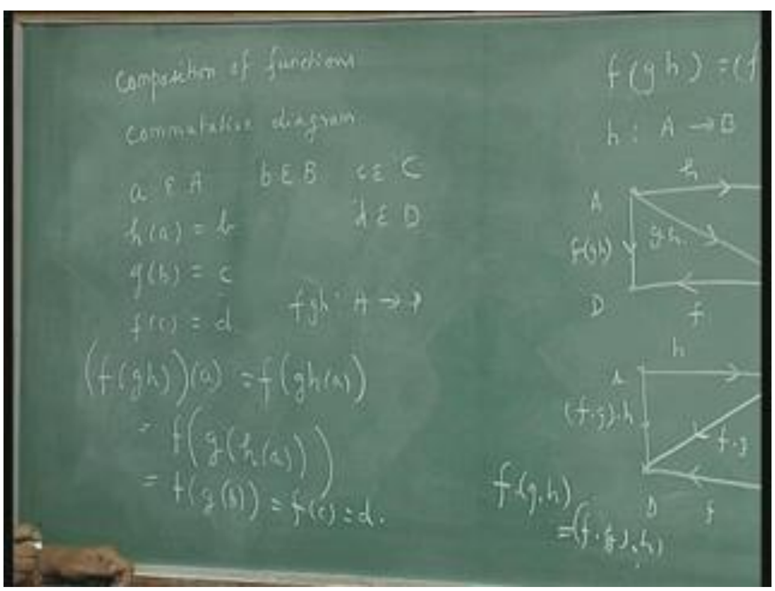
The other way round you can also look at it like this; A B C D this is h and this is g and this is f. So if you combine these two, if you have composition of functions here this will be f dot g and so using this you will get this as f. g and h. So you can see that f dot g dot h is the same as f dot g dot h.

(Refer Slide Time: 23.51)



Or in other words if you have like this; h we have seen that h is from A to B, g is from B to C, f is from C to D. Suppose there is an element a belonging to A such that h of a is equal to b and there is an element c now b belongs to B, c belongs to C, d belongs to D such that h of a is equal to b and g(b) is equal to c and f(c) is equal to d. Then what can you say about (f(gh)) (a)? f(gh) will be from A to D, how can you define that? f(gh) of a will be is equal to f(gh) (a) and what is gh(a)? f(gh) (a) and that is equal to f(g(ha)) is b is equal to f(gb) is c f(c) is equal to d.

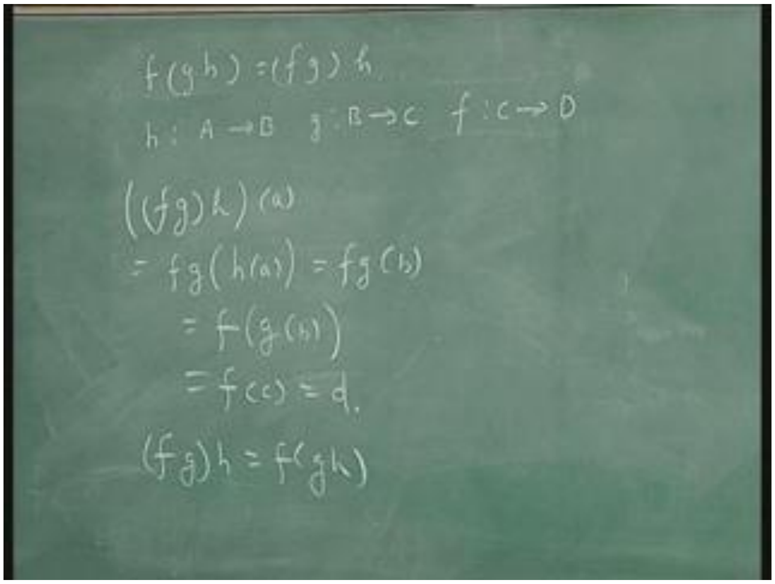
(Refer Slide Time: 25.33)



Now the other way round, so if you consider f(g(h(a))) what will be this?

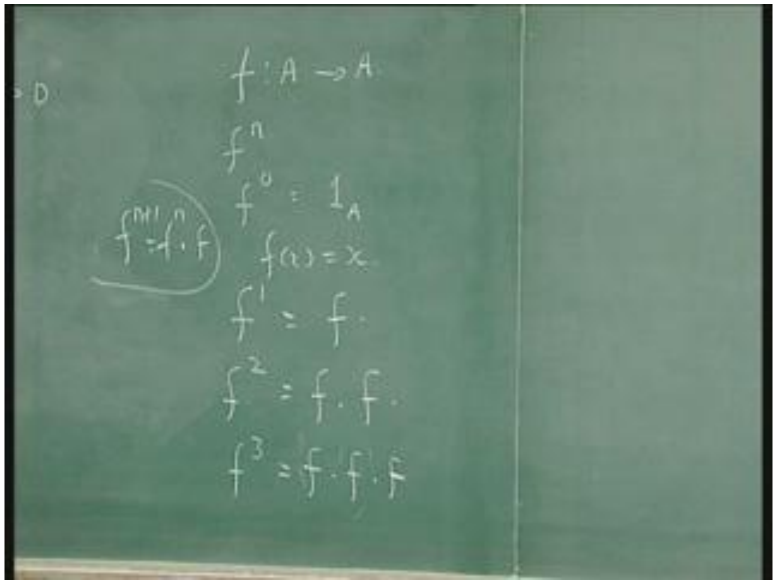
This is f(g(ha)) and we know that ha is b. So this is f(g(b)) and what can you say about this? This is f(g(b)) and what is g of b that is c fc is equal to d. So for every element we can prove like that. So f(gh) is equal to f(gh) or composition of functions is associative.

(Refer Slide Time: 26.16)



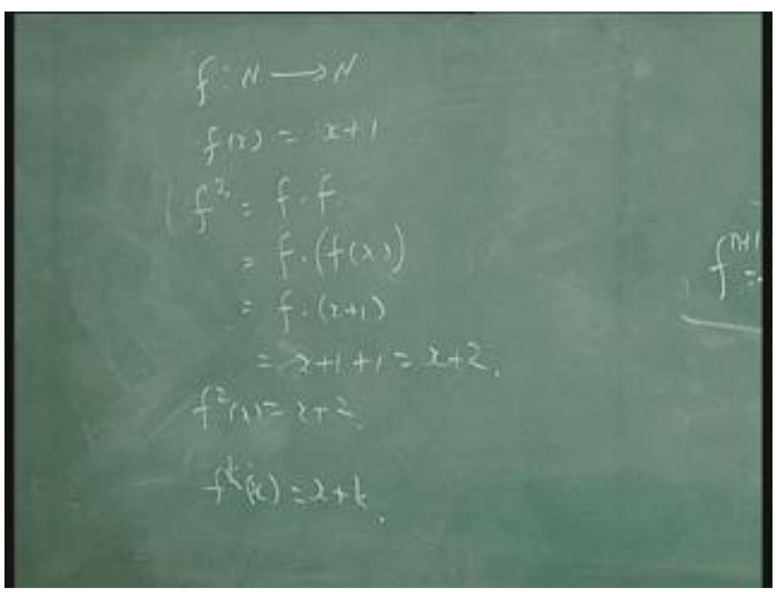
Now because of composition of function is associative without any problem if f is a function from A to A then without any problem you can talk about f power n, f power 0 will be the identity function you denote it like this that is f(x) is equal to x you define like this then f power 1 is f, then f power 2 is f(f) and f power 3 is f cross f cross f because of associativity you can put the bracket this way or this way it does not matter. So without any ambiguity you can talk about f(n), f power n plus 1 will be f of n cross f. And you are able to define this because of the associative property.

(Refer Slide Time: 27.23)



For example, consider f from non negative integers to non negative integers f(x) is equal to x plus 1. Then what can you say about f square? That is f cross f that will be f cross f(x) if you take x as the argument x, f of f of x is x plus 1 so that is x plus 1 plus 1 is equal to x plus 2. So f squared will represent f squared of x will be x plus 2. You can very easily see that if you say that f power kx that will denote x plus k, this is a example where you see the composition of functions.

(Refer Slide Time: 28.27)



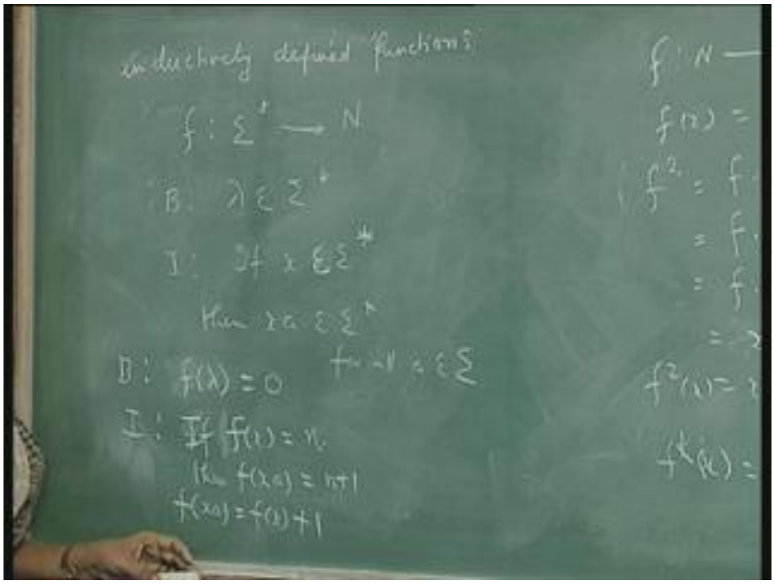
Now you can talk about inductively defined functions. What is this? We have seen that a set can be defined inductively and a relation can be defined inductively. And because functions are particular cases of relations you can also define function in a inductive manner. But when you do that the underlying set, a function is a mapping from a set A to a set B, if the underlying set A is inductively defined then you can use inductive definition for defining a function.

Let us consider some examples. Take the example of a function which maps an alphabet sigma8 into the set of non negative integers. And this function defines the length of a string. Suppose f(x) denotes the length of x, for example suppose I take sigma to be a, b and a, b, c then a string x to be abbacab something like that, then what will be the length of x? The length of x is 1 2 3 4 5 6 7 so 7 so it maps a string into a non-negative integer. This function is defined like this. How do you define this function inductively? How can you define this function in an inductive manner?

First of all you note that sigma power star can be defined inductively. How do you define sigma power star inductively? The basis clause will be lambda belongs to sigma power star. And the induction clause will be if x belongs to sigma power star then xa belongs to sigma power star for all a belonging to sigma. This is how you define this set sigma power star in an inductive manner, lambda is the empty string and x is some string, lambda is the string of length 0 it is a empty string then if x is some string belonging to sigma power star then a is one alphabet then xa belongs to sigma power star for all a belonging to sigma, this is the induction clause. This shows how to build more and more elements of the set from the basis blocks.

Now in a set you also have the extremal class of definition. That is all the elements of the set are formed like this. This is the smallest class which is defined in this manner. We have already seen these things. Now, knowing that this can be defined inductively like this how can you define this function inductively? You see that the basis class you can define like this; the length of lambda, f is the length so f of lambda you can say is 0 the length of the empty string is 0 and the induction portion you define like this; how do you define these strings inductively?
If x belongs to sigma power star, xa belongs to sigma power star. So if f(x) is equal to some n then f(xa) will be n plus 1 this is what we want to say. Or you can say in an easier way f(xa) will be f(x) plus 1 you can define like this. Usually when you define a function you need not have to define the extremal class because that is not necessary. Even though defining the underlying set you need the extremal class. So the basic building blocks are lambda here. So f of lambda is 0 you defined. Then in the induction class from x xa is built. So you see that if f(x) is equal to n f(xa) will be n plus 1 or you can say that f(xa) will be f(x) plus 1 like this you can define a function inductively.

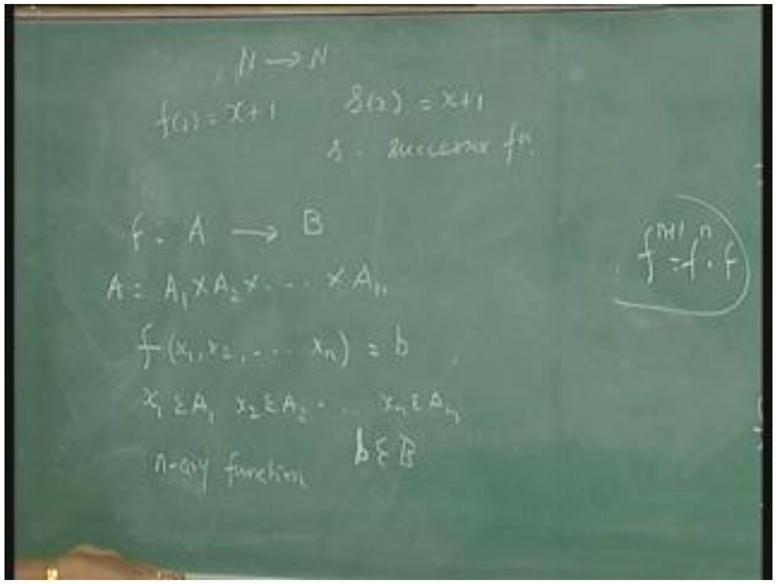
(Refer Slide Time: 33.20)



Let us consider some more examples because this is the way certain things are defined. Take the set of non negative integers to non negative integers. Then x plus 1 instead of saying f(x) is equal to x plus 1 you usually call it as a successor of x, s(x), s(x) is equal to x plus 1 where s is the successor function. Then making use of this you can define addition.

Now, one more point I want to mention is usually we have taken the function from A to B. Now, A could be A_1, A_2, A_n, it could be a Cartesian product of n sets. Then you define the function like this; f is from A is equal to like this. then you say f(x_1, x_2, x_n) is equal to b you define like this where x_1 will belong to A_1, x_2 will belong to A_2 and so on xn will belong to A_n and b will belong to B, this is called an n-ary function. There are n arguments here these n arguments are mapped on to a single argument like that.

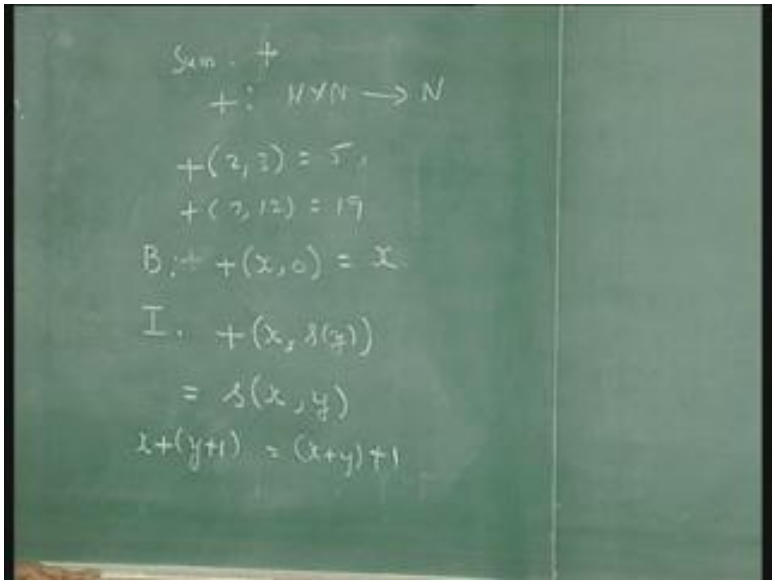
(Refer Slide Time: 35.11)



For example, if you take the sum of two integers 2 plus 3 is equal to 5 like that so this is defined like this, if you denote this by plus where plus maps N cross N to N, let us take the non-negative integers it maps N cross N to N like this. For example, plus of 2, 3 is equal to 5 and plus 7, 12 is equal to 19 and so on.

How do you define this function in an inductive manner?
You can make use of the successor function for this. So plus of basis class you can define like this, plus of a number x with 0 is x itself. If you add 0 to any number you will get x so you can define like this. Now the induction class here will be something like this; plus of x with successor of a number y will be equal to successor of x, y. Or in other words it means that if you want to add x with y plus 1 successor of y will be y plus 1. If you add like this, this is equivalent to saying find the sum of x plus y and find the successor that is this. So this sort of a definition will be used when you define recursive functions.

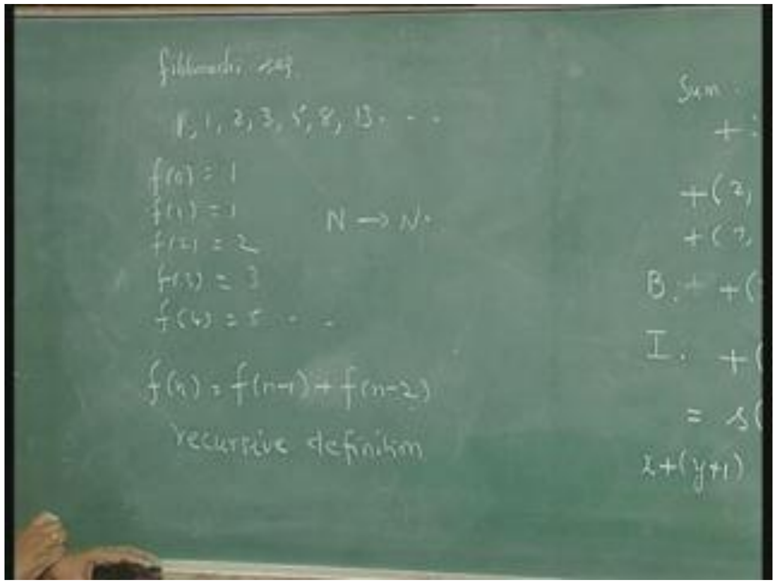
(Refer Slide Time: 37.07)



Recursive functions are defined using some constant functions and composition of operations and what is known as primitive recursion, bounded minimalization and so on, primitive recursive functions these things we shall study later. But this is just to tell you how to define this function in an inductive manner. You can also see that the Fibonacci numbers are defined in this way, another example is like this; consider the Fibonacci numbers. It is 0 1, f(0) is 1, so 1, 1, 2, 3, 5, 8, 13 and so on. That is f(0) is 1, f(1) is 1, f(2) is 2, f(3) is 3, f(4) is 5 and so on.

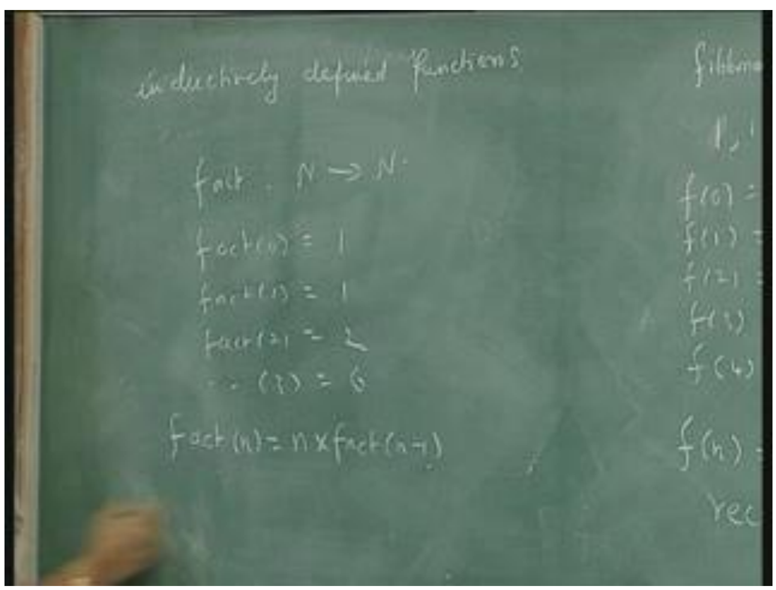
In general, you find that f of n is equal to f(n) minus 1 plus f(n) minus 2. It is from natural non-negative integers to non-negative integers this is known as the Fibonacci sequence. So you see that f(0) is 1, f(1) is 1, 1 plus 1 is equal to 2, 2 plus 1 is equal to 3, 2 plus 3 is equal to 5 and so on. This is also known as a recursive definition of the function.

(Refer Slide Time: 39.12)



Similarly, factorial function if you take, factorial is again from N to N, factorial 0 is 1, factorial 1 is 1, factorial 2 is 2, factorial 3 is 6 and so on. So, in general factorial n is n cross factorial n minus 1 you can define like this. This is again a recursive definition of a function.

(Refer Slide Time: 39.55)

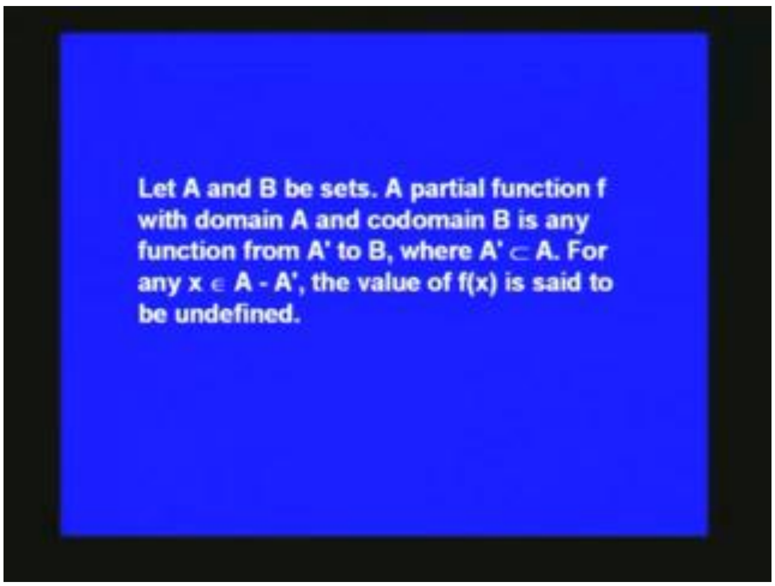


Now, when a function is defined recursively like this to calculate that function sometimes it will be advantages to use a recursive procedure, you know what is meant by an iterative procedure and a recursive procedure.

Factorial can be calculated both by writing a recursive routine and by a procedure which uses iteration. So usually when a function is defined in this manner recursively you can use iteration to write a program to calculate that function or you can use a recursive routine to calculate that function.

Now, sometimes in the definition of the function we have seen that for every element of the domain it has to be defined. You may want to relax a little bit and say that for some elements of the domain it need not be defined then you call it as a partial function. Here again the other condition that each element should be mapped on to a unique element of B is there. That is you cannot map one element of A into two different elements of B. But for some elements of A you need not define the function, you need not have the map that is known as a partial function.

(Refer Slide Time: 41.31)



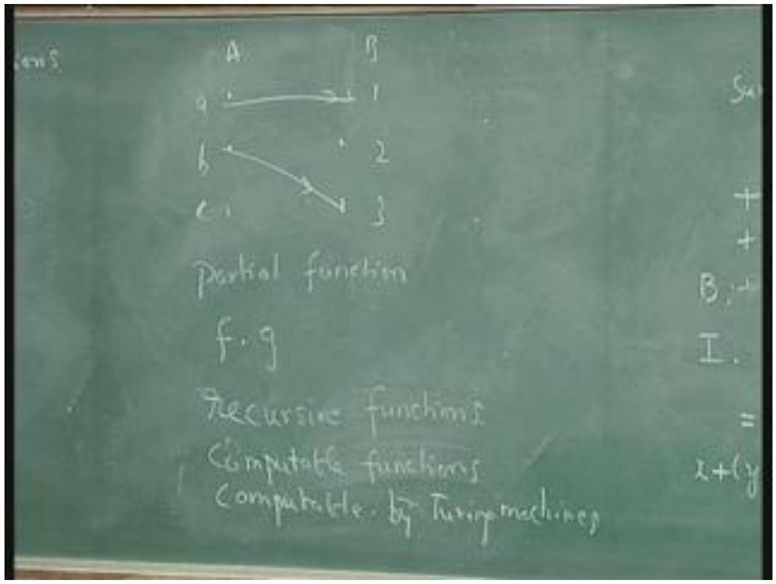
Let A and B be sets. A partial function f with domain A and a codomain B is any function A dash to B where A dash is a subset of A. For any x belonging to A minus A dash the value of f(x) is said to be undefined.

For example, take this function, take a b c 1 2 3 A to B the function is from A to B defined in this manner f of a is 1, f of b is 3 but f of c is not defined, so when we consider the function initially we said this is not a function, this is not a total function but it is called a partial function because for c it is not defined but for a it is 1 and for b the map is 3 so this is called a partial function. Again partial functions also we can define composition of functions. When you have two partial functions f and g you can talk about the composite function f. g that will also be a partial function. Then the composition will be associative and other properties like that you can define.

In general, the recursive functions form an important class. We shall learn about that a little bit later. We talked about total recursive functions and partial recursive functions.

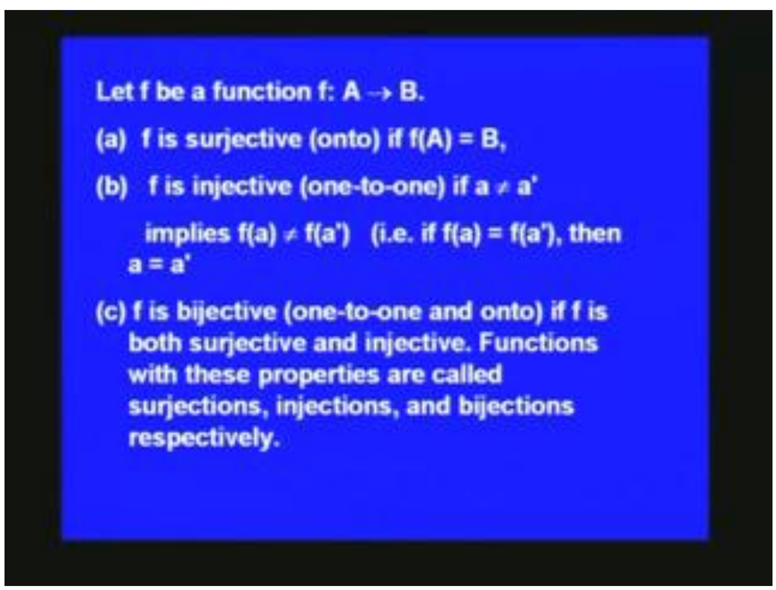
Partial recursive functions are the functions which are exactly computed by tuning machines. And it is the class of computable functions. They are known as computable functions. That is, they are computable by what is known as a tuning machine.

(Refer Slide Time: 43.55)



Now, we will consider some special properties of functions.

(Refer Slide Time: 44.18)



Let f be a function f from A to B. f is surjective or onto if f(A) is equal to B. f is injective one-to-one if a dash a not equal to a dash implies f(a) not equal to f(a dash) that is if f(a) is equal to f(a dash) then a is equal to a dash. f is bijective one-to-one and onto if f is both surjective and injective functions with these properties are called surjections, injections and bijections respectively. So let us consider some examples first with finite sets and other sets you can consider.

Look at this, 1 2 3 from A to B this is a and b this is the set B. Now 1 is mapped onto a, 2 is mapped onto a, 3 is mapped onto b. Now, this is the domain, this is the codomain. Every element of the codomain is the image of some element of an element here. And such a function is called onto function or a surjective function.

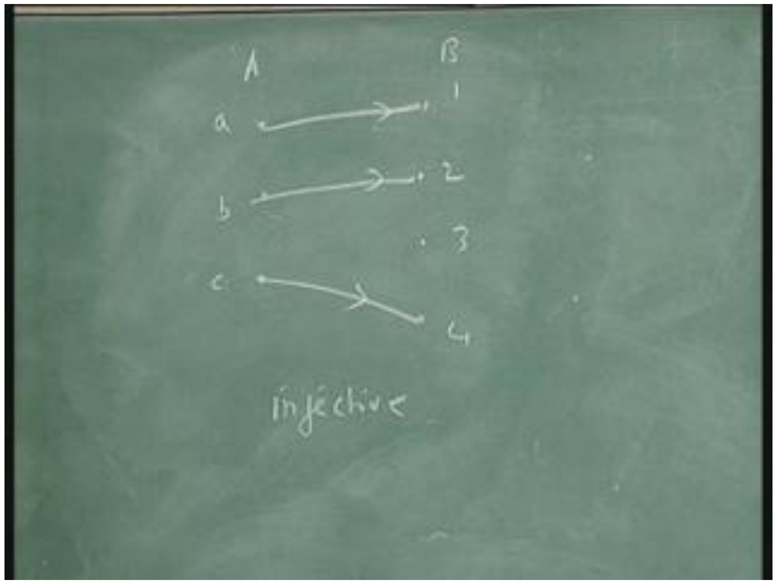
Look at this function, from a b c to d e f where a is mapped onto e, b is mapped onto e, c is mapped onto f. This is a function no doubt but d is not the image of any element here so this is not a surjective function. This is an example of a surjective function, this is an example of a function which is not surjective because d is not the image of any function.

(Refer Slide Time: 46.38)

So look at the first definition; f is surjective or onto if f of A is equal to B. Second is, if f is injective one-to-one if a is not equal to a dash implies f(a) not equal to a dash.

Again let us consider some examples here, you have A and you have B suppose I have a b c 1 2 3 4 a is mapped onto b, b is mapped onto 2, c is mapped onto 4. This is an example of an injective function because different elements should be mapped onto different elements. In that case you say that the function is injective or one-to-one.

(Refer Slide Time: 47.36)

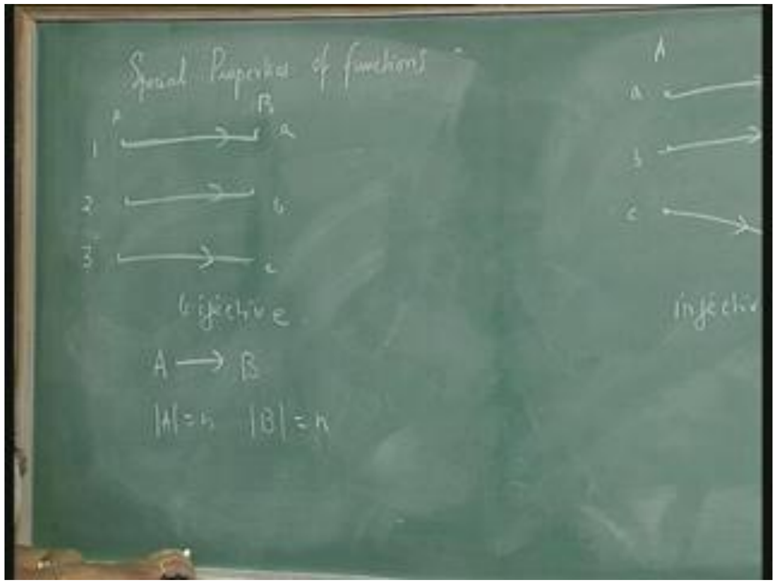


Look at this example, here 1 and 2 are mapped onto the same element so this is not an injective function. It is not an injective function because two elements are mapped onto the same element. If f(a) is equal to f(b) then a should be is equal to b here f1 and f2 are equal but 1 and 2 are not equal, so this is an example of a function which is not injective and this is an example of a function which is injective. If a function is injective as well as surjective it is called a bijective function.

A bijective function will be like this, look at this function here, 1 is mapped onto a, 2 is mapped onto b, 3 is mapped onto c so the whole set is the image of the function and every element here is the image of some element here so it is surjective, different elements are mapped onto different elements so it is injective so this is an example of a bijective function, it is both injective and surjective.

Now you must note that if you are having finite sets, a function defined between functions A and B are finite set if A has n elements and if it is a bijective function both of them should have the equal number of elements then only different elements can be mapped onto different elements and the whole set B will also be covered. So if it is on finite sets A and B you will have the same number of elements in A and B.

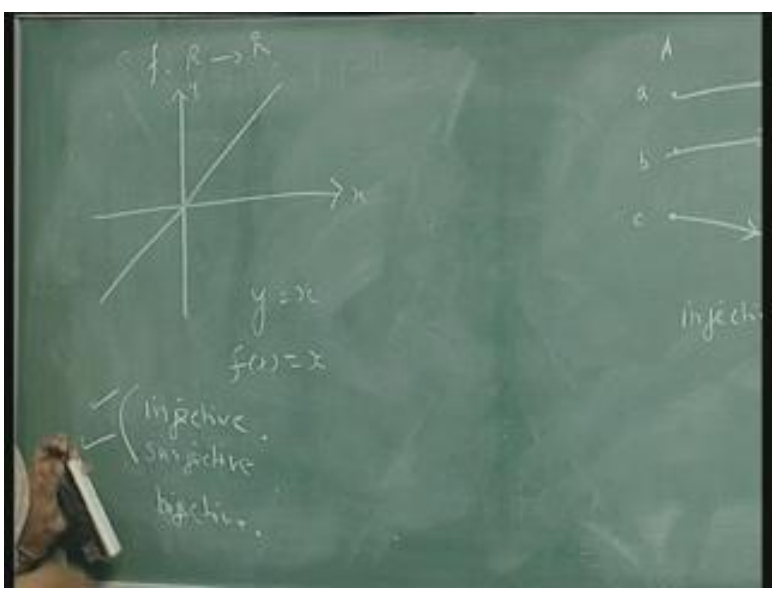
(Refer Slide Time: 49.29)



Now let us consider some real functions. We will see what happens, set of functions functions from real number to real number.

Look at this line, this is the x axis this is the y axis you define the function y is equal to f of x. Now this denotes y is equal to x like line or here the function is f of x is equal to x. What can you say about this function? This is, different elements are mapped onto different elements so it is injective. and every element is the image of some element, if you take some number y y will be the image or if you take some real number p p is the image of p so it is also surjective and hence because these two are satisfied it is bijective.

(Refer Slide Time: 50.40)

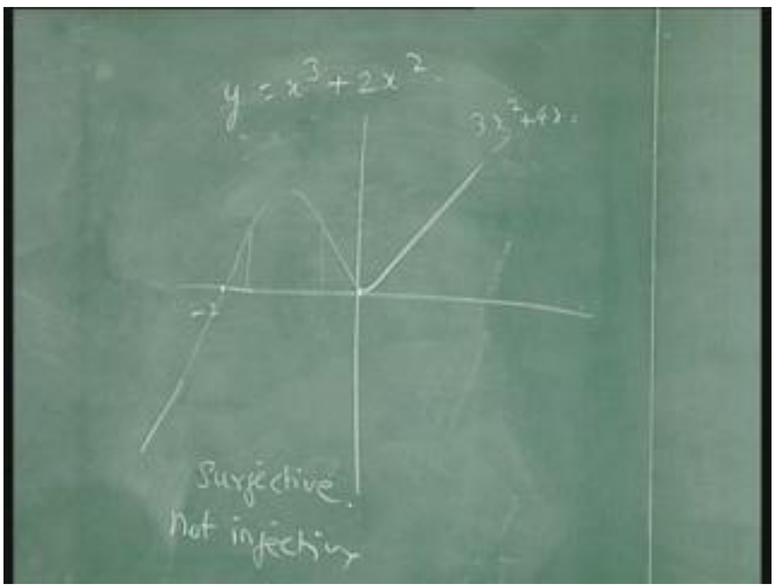


Now, look at the function y is equal to e power x, the graph will be like this what can you say about this? Different elements will be mapped onto different elements, you can see that if you have x1 not equal to x2 e power x1 will not be equal to e power x2 and so different elements will be mapped onto different elements so it is injective.

But the negative numbers are not the image of any elements here so it is not surjective. This is an example of a function which is injective but not surjective.

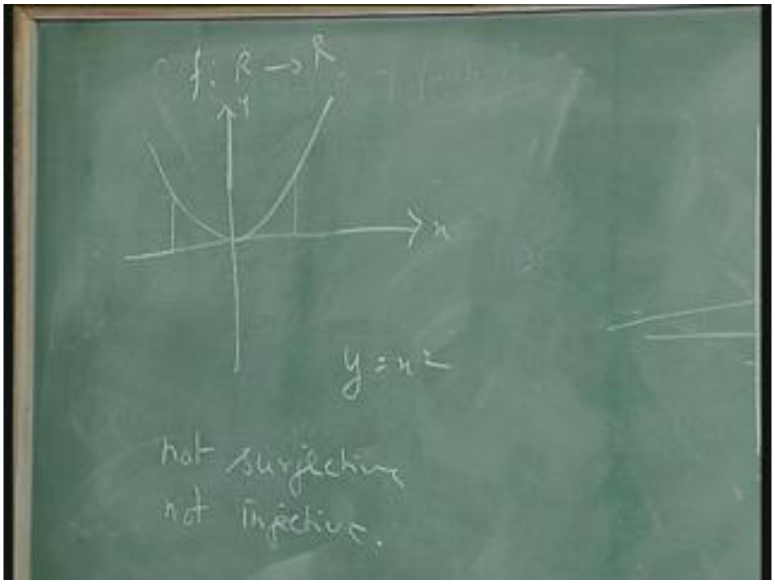
Look at this function y is equal to x cubed plus 2x square something like that, how will the graph look like? It is 3x square plus 4x would be 0 x is equal to 0 and then where will you cut the x axis, it is at minus 2x is equal to minus 2 so the graph will be something like this. Is this a surjective function? You can see that for every element there will be one element for which it is the image so it is an example of a surjective function. But here two elements are mapped onto the same element so it is not injective. This is an example of a function which is surjective but not injective.

(Refer Slide Time: 52.43)



And look at this function y is equal to x square the graph will look like this. What can you say about this function?
The negative numbers are not the image of any elements so you will find that it is not surjective and two different elements will be mapped onto the same element here so it is not injective. This is an example of a function which is neither surjective nor injective.

(Refer Slide Time: 53.24)



So it not necessary that all functions should be injective or surjective or something like this. Some functions can be injective functions, some functions can be surjective functions, some functions will be bijective functions and so on.

Why should we consider this? What is the necessity to consider these properties of such functions? There are some functions known as the hash functions, some functions are called hash functions which are very useful in Computer Science and in compilers. For that you would rather prefer to have a bijective function if not you will at least try to have a function which is close to an injective function and so on.

We shall see this application in the next lecture.