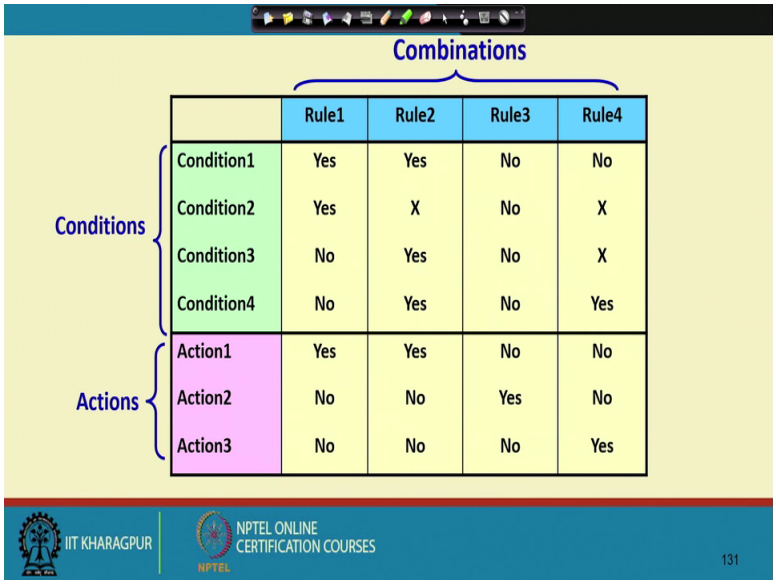
Software Engineering
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 51
Combinatorial Testing

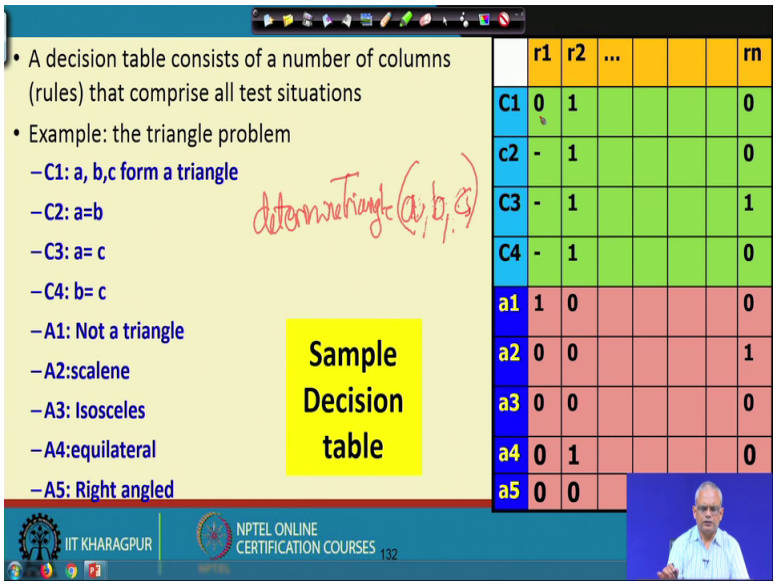
Welcome to this lecture. In the last lecture, we were discussing about the Combinatorial Testing and, as the first testing technique the combinatorial testing we are discussing about the decision table based testing. We had said that if there is a function, whose output depends on different types of conditions, then we can represent the values of the conditional inputs and the corresponding actions, in the form of a decision table. And, every column in the decision table become a test case and the we call as a rule.

(Refer Slide Time: 01:04)



So, suggest to refresh what we were discussing last time. We can represent the outcome of different conditions or the values of different condition the conditions. And, the corresponding actions let us say 3 actions and sometime depending on the values of the condition, the action may take place or does not take place. This forms our decision table and each of the rule will become a test case. And for some of the values of the conditions, we have written do not care. So, in that case irrespective of the condition 2 value the rule 2 will result in this action.

(Refer Slide Time: 02:05)

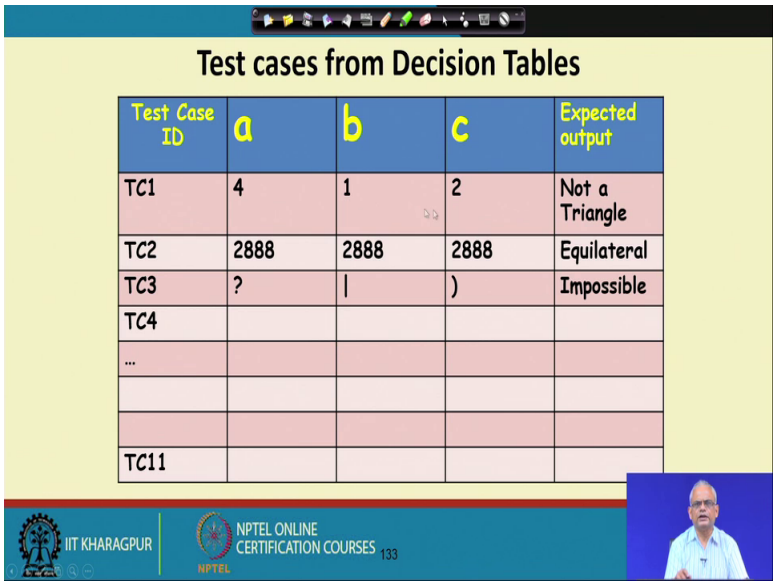


Now, let us do some example problems first we start with something very simple. Again, we take the problem of a function, which takes 3 parameters a b c representing the sides of a triangle. Determine triangle type and we have 3 parameters side 1, side 2, and side 3, these are the 3 parameters for the determine triangle function. And, then depending on the specific values of the 3 sides it can display not a triangle, scalene, isosceles, equilateral, right angled etcetera.

Now, the conditions here are that a b c form a triangle or C 2 is a equal to b, C 3 is a equal to c, C 4 is b equal to c ok. The parameters let me just rewrite them as a b c that maybe easier to understand in this context. So, the parameters are a b c these are the 3 sides, a equal to b a equal to c b equal to c these are the 3 conditions in the parameters. And, now we represent the, it in a decision table form. If C 1 is a 0 that is it does not form a triangle then we have a 1.

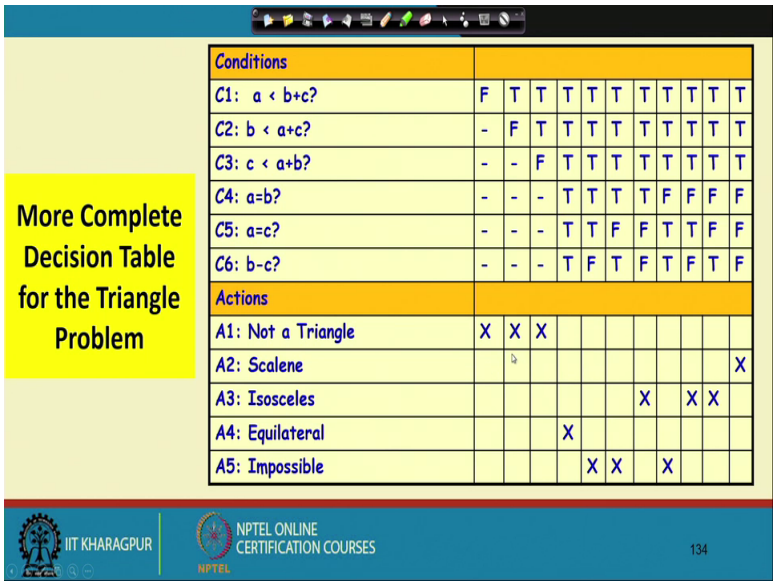
Now, if it forms a triangle and then C 2 C 3 C 1 are 1 then it is equilateral all sides are equal and, similarly we just keep on filling this and, then each column we make a test case. So, that is the essential idea behind the decision table testing, whenever we see that some relation between the input parameter help define the output parameter. We write that in the form of conditions and those conditions, we represent here all possible values of conditions and then we note the corresponding actions and then these form our test cases.

(Refer Slide Time: 05:04)



And, we can write the specific values for the parameter a b c for which the columns that define the test case. For example, a is not equal to b, b is not equal to c and c is not equal to a, which is not ok, not a triangle we can take 4 1 2 as a specific values here and so on.

(Refer Slide Time: 05:44)



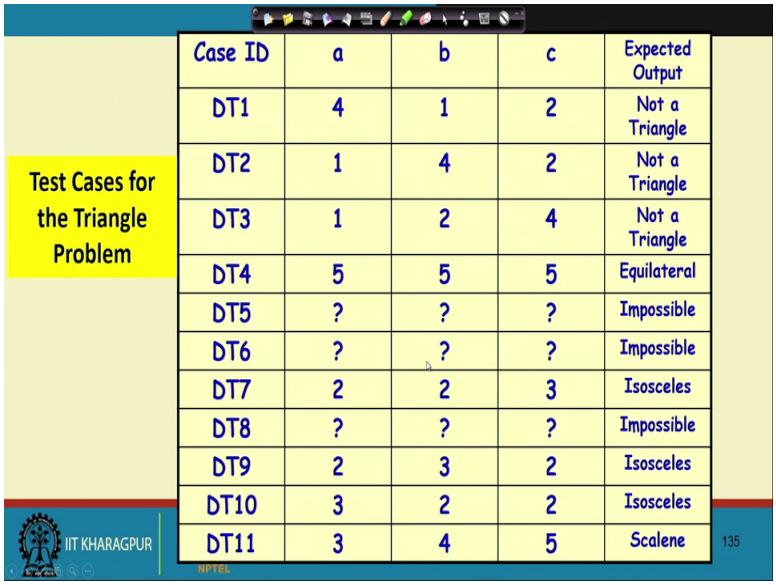
But, this is a more complete decision table, because earlier we had one of the condition is not a triangle, but what are the conditions, which define not a triangle?

The conditions are if either if none of these are true, then this becomes not a triangle. If either of these becomes true that a is greater than b plus c or b greater than a plus c and

so on. Then, it is not a triangle and we represent this C 1 C 2 C 3, which help us to form a more concrete decision table and this will become easier to translate into test cases.

If any of this is false C 1 C 2 C 3 the corresponding, which as straight a not a triangle so, compared to our previous decision table this is a more complete decision table, because it helps us to easily define the test cases.

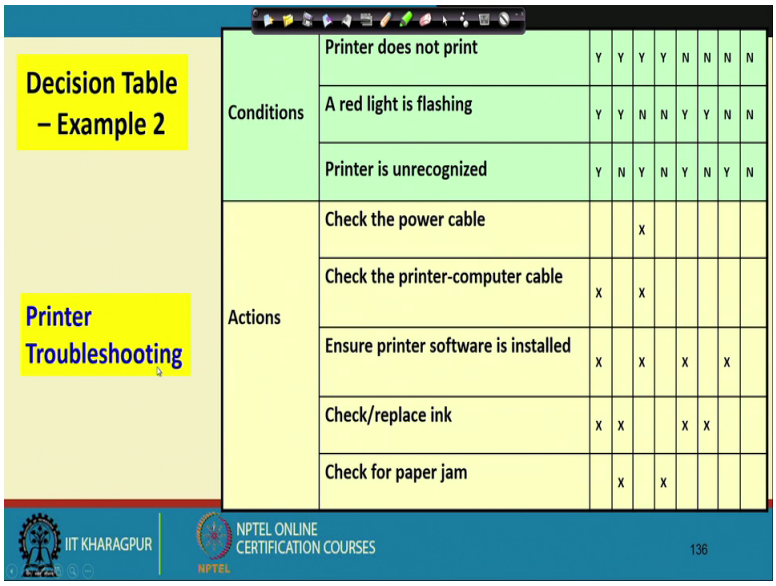
(Refer Slide Time: 07:05)



Case ID	a	b	c	Expected Output
DT1	4	1	2	Not a Triangle
DT2	1	4	2	Not a Triangle
DT3	1	2	4	Not a Triangle
DT4	5	5	5	Equilateral
DT5	?	?	?	Impossible
DT6	?	?	?	Impossible
DT7	2	2	3	Isosceles
DT8	?	?	?	Impossible
DT9	2	3	2	Isosceles
DT10	3	2	2	Isosceles
DT11	3	4	5	Scalene

And, based on the decision table we can form the test cases we can give the values. And then what is the expected output right and it each of these is test case and we get 11 test cases.

(Refer Slide Time: 07:22)

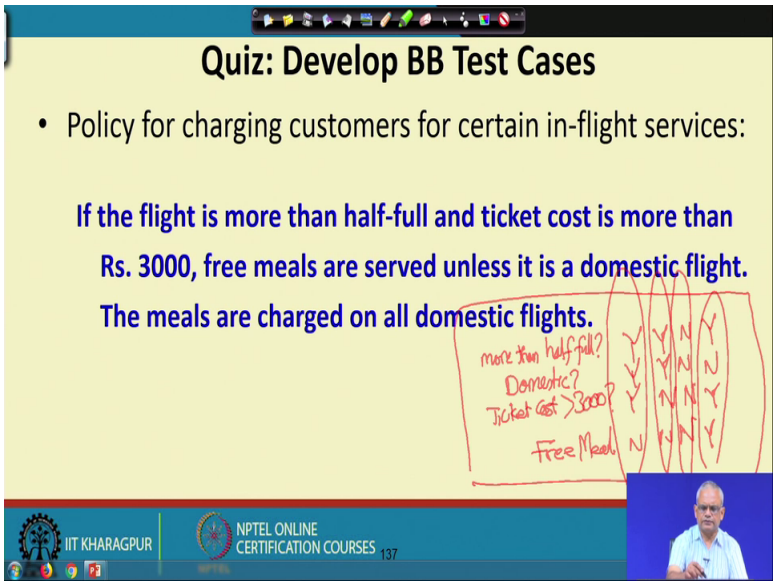


Now, let us take another decision table example. Let us say the function is printer troubleshooting. We, give some input to the function those are the conditions for example, if the printer does not print. Red light is flashing and printer unrecognized, based on the logical combinations of these input the output is defined.

For example, if the printer does not print red light is flashing. And, printer is unrecognized, then the output will be check the printer computer cable and see your printer software is installed and check or replace the ink, but if the printer does not print and the red light is flashing. Then, we will display that check and replace ink and or check for the paper jam and so on.

For every combination of the input parameter, we write the corresponding actions. And, this forms the decision table for the printer troubleshooting. And, then once we develop this decision table each of this row sorry each of the column in the decision table forms a test case.

(Refer Slide Time: 09:00)



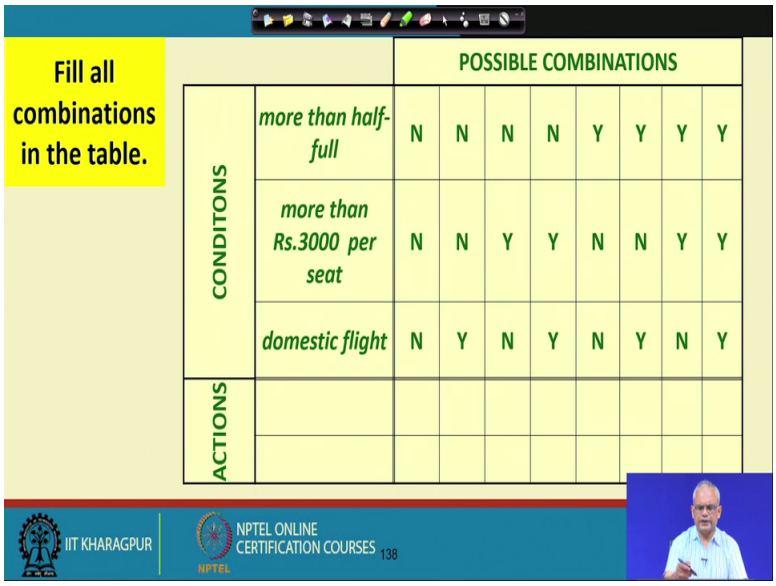
Now, based on our understanding of the decision table, let us try to solve a very simple problem. Let us say certain airline provides in-flight services that is food free meals are served if the fight is more than half-full and the ticket cost is more than 3000.

If, the flight is more than half full and ticket cost is more than 3000, free meals are served unless it is a domestic flight. If it is a domestic flight, then meals will not be served. And, the meals are charged on all domestic flight. So, how do we develop the decision table for this? If we look at the conditions the conditions are whether more than half full, it is a domestic or international flight and ticket cost greater than 3000 these are the 3 conditions. And the action is free meal served or not.

Now, we can form all possible combinations of this more than half full, yes domestic, yes ticket cost 3000, yes free meal no. More than half full yes, domestic yes, ticket cost greater than 3000 no, and in that case free meal is no. More than half full no, domestic no, ticket cost greater than 3000 no, and then no free meal is served, but what about more than half filled is yes domestic is no ticket cost is greater than 3000, then it becomes yes and so on. So, we can develop the set of test cases for this by looking at the rules.

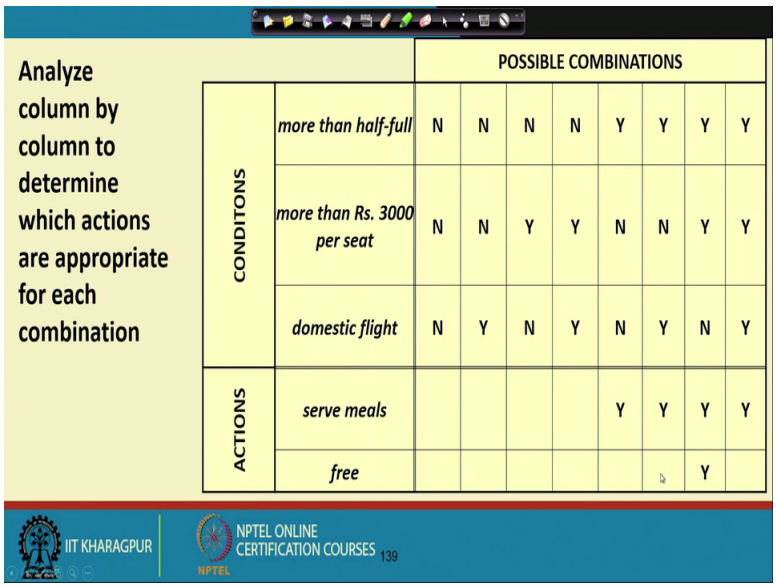
But, how do we optimize the number of test cases? Let us look at that.

(Refer Slide Time: 11:53)



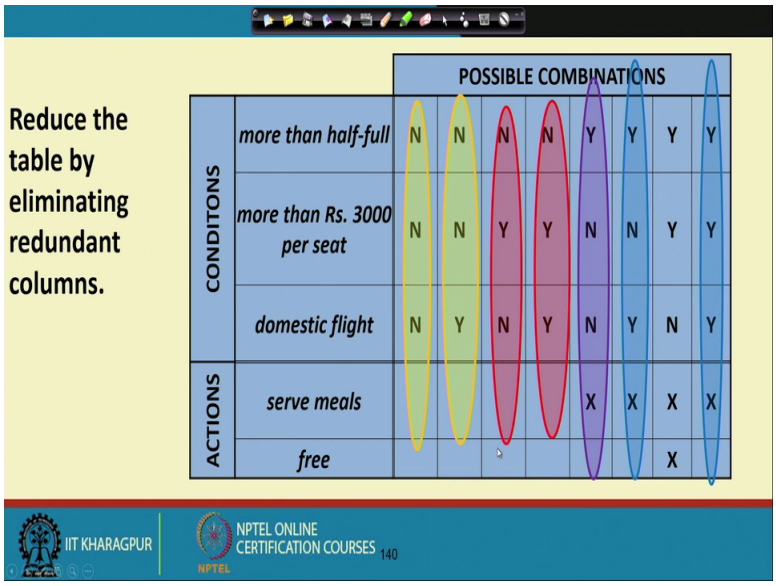
So, we can identify the 3 conditions more than half full, more than 3000 per seat and the domestic flight and then the action is free meals are charged. First is we form all possible combinations each one has a 2 values yes and no. So, 2 to the power 3, we will need 8 of them here.

(Refer Slide Time: 12:31)



So, we can represent now the corresponding actions. As long as it is a more than half full, meals are served and if it is a international flight it is not a domestic flight, then it is a free meal.

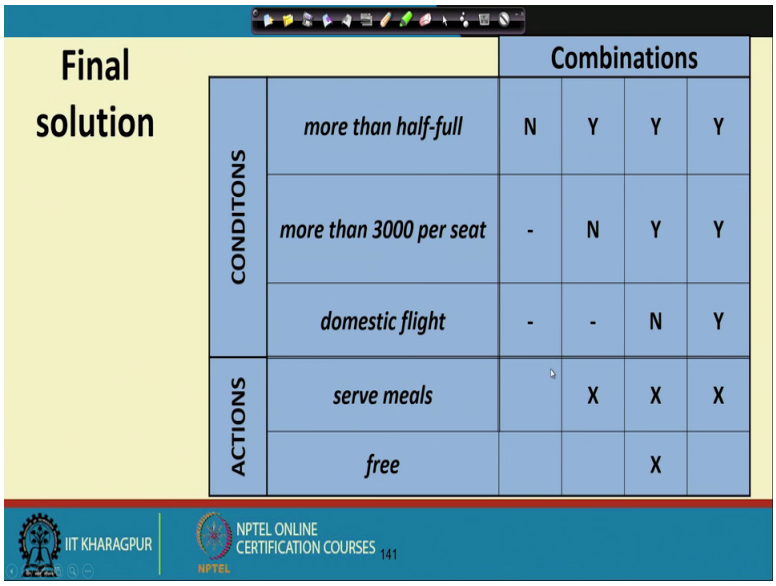
(Refer Slide Time: 13:09)



But, one thing we can observe here is that if we represent it in the form of a decision table. And, we will look at the actions for this both these rules the action is same and the only difference between the conditions here is that this is No and this is Y. So, we can combine these 2 we can combine these 2 and we can write do not care for this. Similarly for these 2 they are the rules are similar, I mean the conditions are similar accepting that these 2 are different rest are similar and also the action is the same. And therefore, we can combine these 2 and write do not care for this.

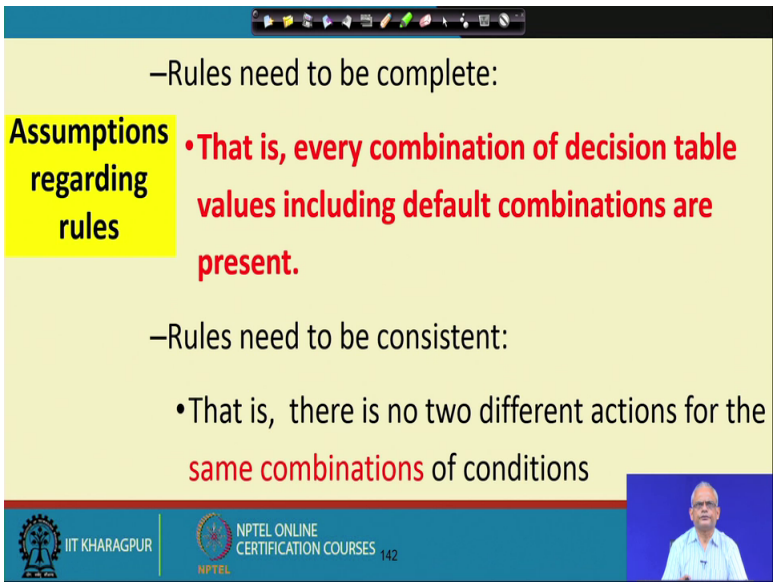
And, similarly you can combine these 2 because they differ only with respect to more than 3000 per seat, but the action is the same they serve meals. Now based on this idea we can combine this and form a decision table.

(Refer Slide Time: 14:32)



Which is more compact number of rules are less, we have use the do not care here. And therefore, the number of test cases is reduced to 4.

(Refer Slide Time: 14:54)

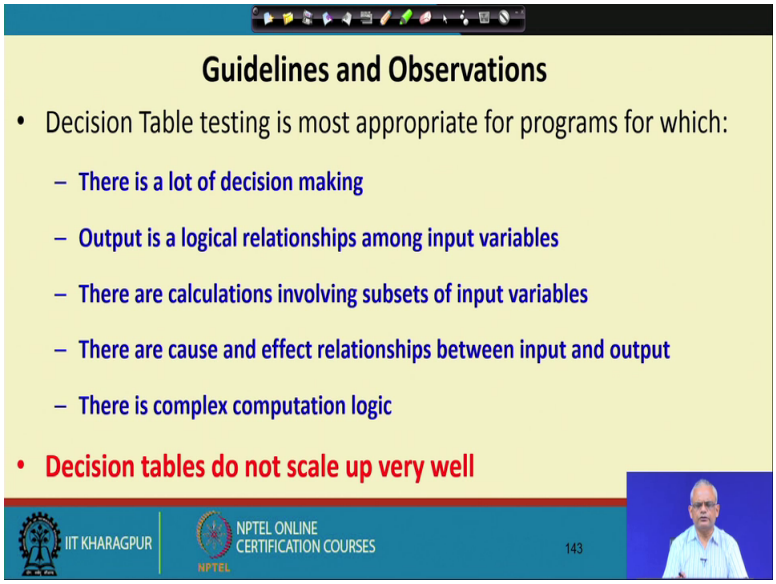


Now, let us see some assumptions made regarding the rules, one is that when you form the decision table all possible combinations of the conditions are represented, that is how we must develop the decision table. So, that we do not miss out on specific combinations. And, one thing we must guard against is that for the same combination of

conditions, same values of the conditions, we cannot have 2 different columns and have 2 different actions taken taking that will be a contradictory thing.

So, we should guard against this while developing a decision table.

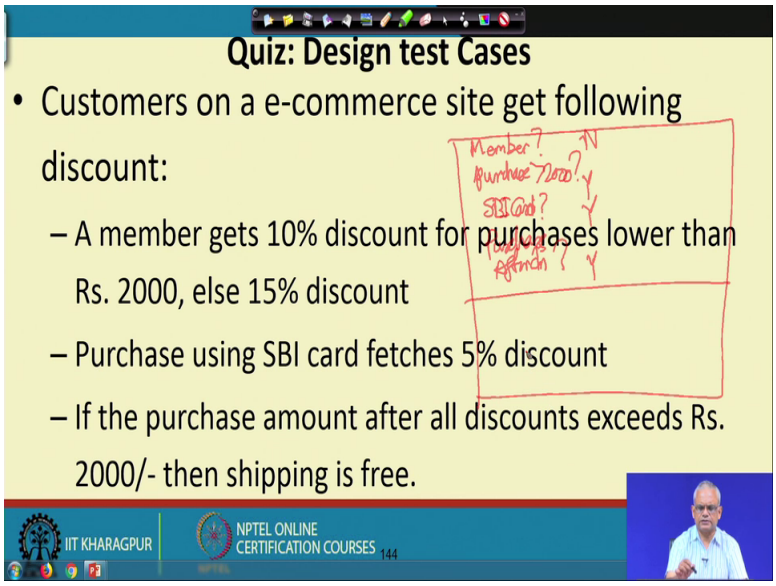
(Refer Slide Time: 15:53)



The problems for which the decision table based testing is applicable is that from the function it appears that there is lot of decision making, if then else kind of thing happens. The output is a logical relation among inputs or there is a calculation involving subset of input variables or there is a cause and effect relation between input and output or the computation logic is complex.

In all these cases we need to develop the decision table, but one thing we must understand that, developing the decision table when the number of conditions is large becomes very cumbersome can make mistakes. So, for small problems involving few conditions decision table is a very helpful to design the test cases.

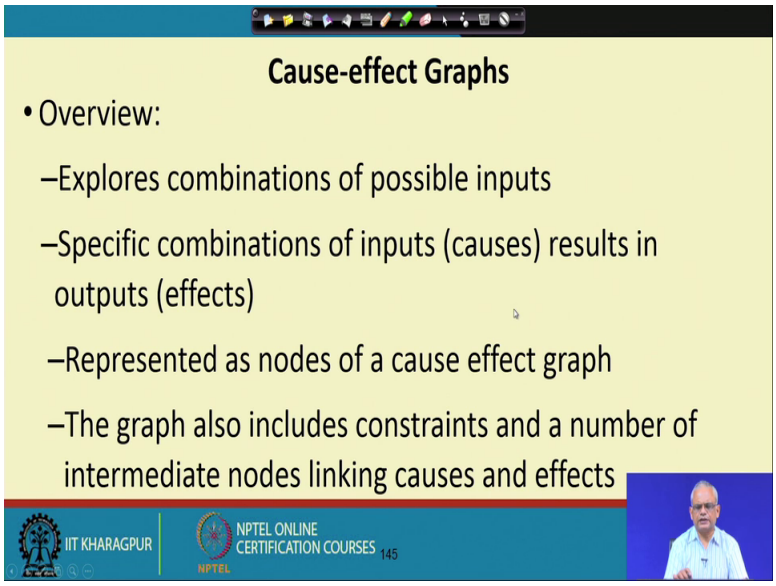
(Refer Slide Time: 17:01)



Now, let us do a small quiz to design the decision table based testing. Let us say the customer on a e-commerce site gets the following discount. A member of the site gets 10 percent discount for purchases lower than 2000, if he is not a member he gets. If the purchase is more than 2000 gets 15 percent discount, purchase using the SBI card fetches 5 percent discount. If the purchase amount after all discount exceeds 2000, then shipping is free how do we design the decision table for this? The first step is to identify the conditions.

The first condition is that is the customer a member. The second condition is that the purchase amount greater than 2000 or not. The third condition is that whether SBI card is used or not. And, then purchase amount after discount. So, these are the 4 conditions and then we can develop the decision table here. Member is no and let us say purchase using yes, greater than 2000 yes, SBI card yes, and the total amount after discount is greater than 2000 yes, then the total discount will be 5 percent for SBI card and then shipping is free and so on. We can design the decision table and that will help us to generate the test cases.

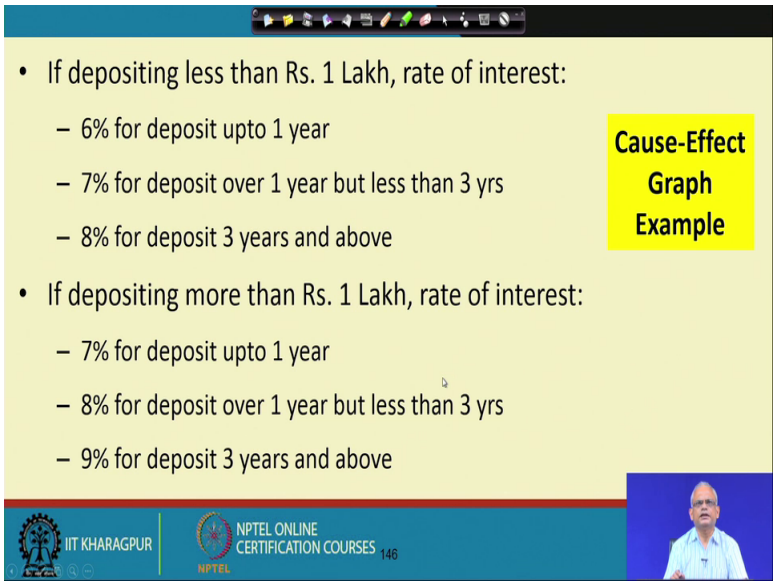
(Refer Slide Time: 19:55)



The cause effect graphs is a testing technique, which just systematizes the decision table development. If we understand the problem well, we can easily generate the decision table, but sometimes the cause effect graphs can help us generate the decision table easily. Here we have specific symbols for input and then we will represent that, the symbols and form of output. And, then we develop the cause effect graph and then by looking at the cause effect graph it becomes easy to generate the decision table.

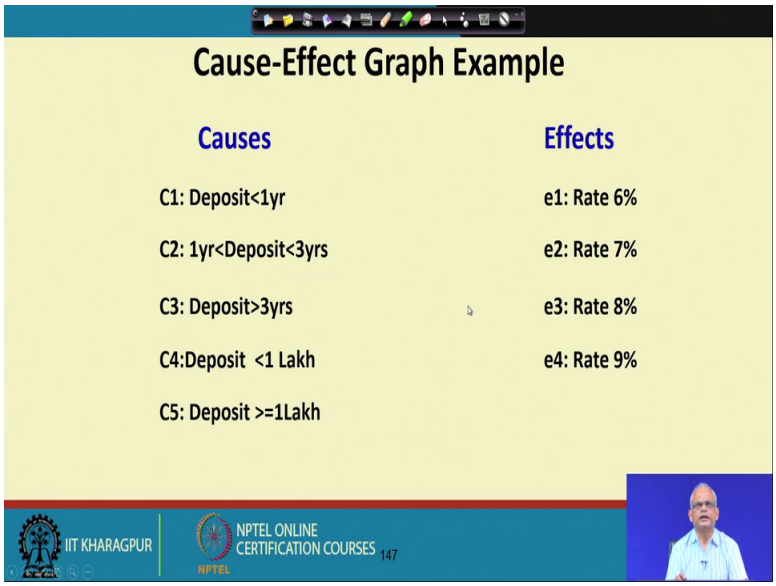
Let us look at one example and based on that we can see that cause effect graph is just a technique, which helps us to develop the decision table. And finally, the test cases are generated based on the decision table, but if we are able to generate the decision table without using the cause effect graph then well and good, but if we cannot really design the decision table, then we can use the cause effect graph represent the causes and effects in the form of a graph. And from that graph it becomes very easy to generate the decision table.

(Refer Slide Time: 21:32)



Let us explain this technique using a simple example again we will take the case of depositing less than 1 lakh rate of interest is 6 percent for 1 year 7 percent for deposit 1 year to 3 year and 8 percent for 3 year and above. If, deposit is more than 1 lakh then 7 percent for up to 1 year 8 percent for deposit over 1 year, but less than 3 year and 9 percent for deposits above 3 years.

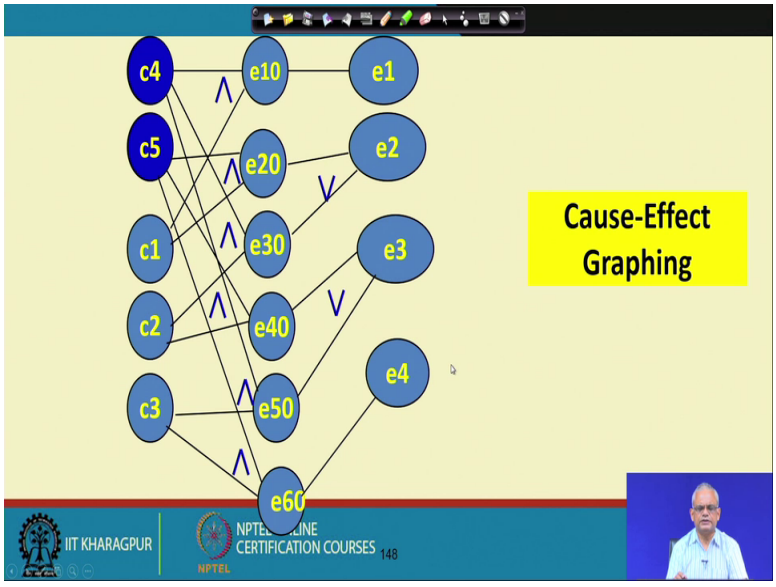
(Refer Slide Time: 22:11)



Now, we can write the causes which are the conditions basically and the corresponding effects. So, deposit is less than 1 year deposit is between 1 to 3 year deposit is greater than 3 year, deposit is less than 1 lakh and deposit is greater than 1 lakh.

So, these are the different conditions or causes. And, then we will have the corresponding effect whether rate is 6 percent, 7 percent, 8 percent or 9 percent.

(Refer Slide Time: 22:48)



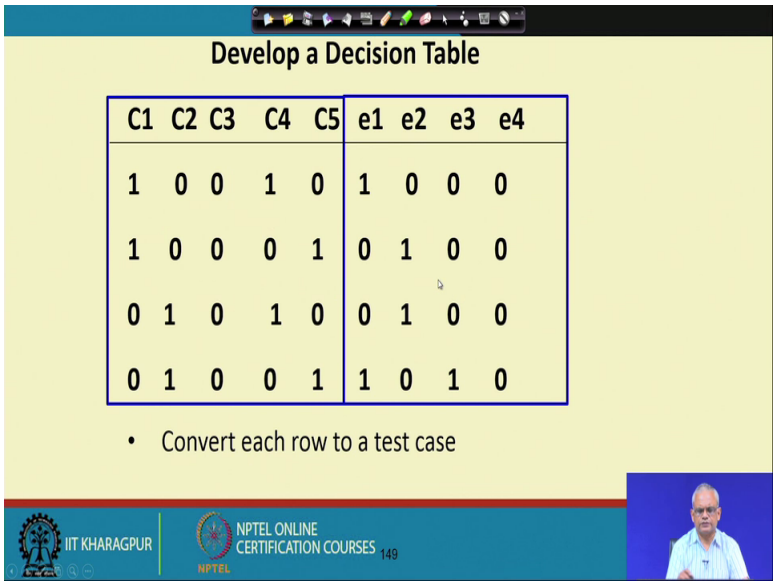
Now, we present in the form of a Graph.

If, you look here that C 1 C 2 C 3 are the duration of the deposit C 4 C 5 are the amount of deposit, whether it is greater than 1 lakh or less than 1 lakh, c 4 c 5 this is whether the deposit is greater than 1 lakh or less than 1 lakh. And, c 1 c 2 c 3 are the duration. Now, if the deposit is less than 1 year and the amount deposited is less than 1 lakh, then the rate of interest is 6 percent. If, the deposit is for more than 1 year and the amount deposited is more than sorry less than 1 lakh then it is 7 percent and so on

So, this is the end condition I have represented here. To indicate the fact that if both of these hold, then this action takes place. And, here for this we have a or condition represented here, because if any of this holds for any of this action takes place. If we have a and here then both of these takes place then this will hold, but we have used or here.

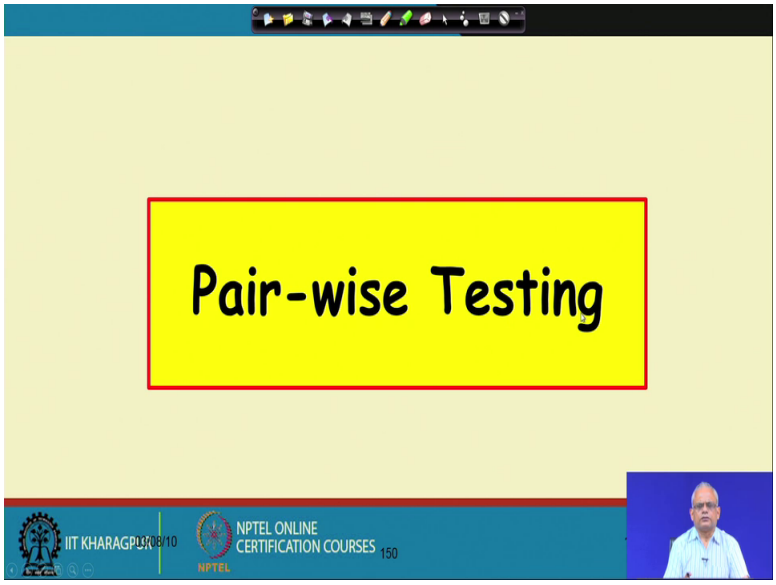
So, cause effect graph is a very simple technique, which kind of helps in developing a decision table. Once we have got this, then developing the decision table become straight forward, we form these conditions and these are the actions and then we look at this and form the different condition values of the conditions.

(Refer Slide Time: 25:09)



So, this is the representation of decision table. These are the actions, these are the conditions and different combinations of the conditions are easily identified from the cause effect graph.

(Refer Slide Time: 25:24)



In the combinatorial testing so, far we have discussed about the decision table based testing. And, we also saw that the cause effect graphing is a systematic technique, which let us develop a graph from the input problem description. And, the graph then can be easily translated into a decision table. And, that helps us to generate the test cases, because every row sorry every column will become a test case.

In our subsequent discussion, we look at the pair wise testing. Sometimes the number of conditions and the actions are too many. We were just discussing about the case of a equivalence partitioning and the boundary value testing. The numbers of parameters are too many, which normally occurs in the case of user interface and controller type of programs, then the number of test cases can become too many. In that case is there any way we can reduce the number of test cases.

But, still achieve as much effective testing as either a robust testing or let us say decision based testing, we will discuss the pair wise testing, which will help us to reduce the number of test cases substantially and still achieve good testing. So, that topic we will discuss in the next lecture, which is pair wise testing.

Thank you.