UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Assignment 2nd

Student Name: Rishav Kumar **UID: 22MCC20039**
Branch: MCA - CCD **Section/Group: 22MCD1 / Grp A**
Semester: II **Date of Performance: 15th April 2023**
Subject Name: Software Testing **Subject Code: 22CAH-685**

Que 1) What is cause-effect graphing technique? Elaborate with example.

Answer)

Cause Effect Graph is a popular black box testing technique. It illustrates the relationship between a given outcome and all the factors that influence the outcome graphically. Here, a "Cause" stands for a distinct input condition that fetches about an internal change in the system and an "Effect" represents an output condition, a system state that results from a combination of causes.

Its applications include: -

1. For analyzing the existing problem so that corrective actions can be taken at the earliest.
2. For relating the interactions of the system with the factors affecting a particular process.
3. For identifying the possible root causes, reasons for a particular effect, problem or outcome.

Advantages of cause-effect graphs are: -

1. It helps to determine the root causes of a problem or quality.
2. It indicates possible causes of variation in a process.
3. It identifies those areas where data should be collected for further study.
4. It utilizes the team knowledge of the process by encouraging team participation.

Example:

There are two columns: col1 and col2. The characters allowed in col1 are 'a' or 'b'. The characters allowed in col2 are digits 0-9. A file is updated if both the columns i.e. col1 and col2 are correct. If col1 is incorrect, message 'X' is displayed and if col2 is incorrect, message 'Y' is displayed. Draw the cause-effect graph for the given problem.

1. Identify the causes:

cl : character in coll is 'a'

c2 : character in coll is 'b'

c3 : character in c012 is a digit

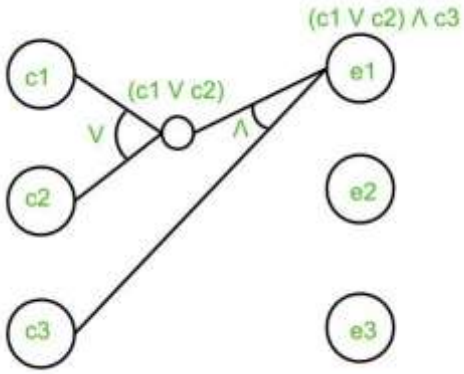
2. Identify the effects:

e1 : file is updated if $((c1 \vee c2) \wedge c3)$
e2 : message 'X' is displayed if $(\overline{c1} \wedge \overline{c2})$ or $\overline{(c1 \vee c2)}$
e3 : message 'Y' is displayed if $\overline{(c3)}$

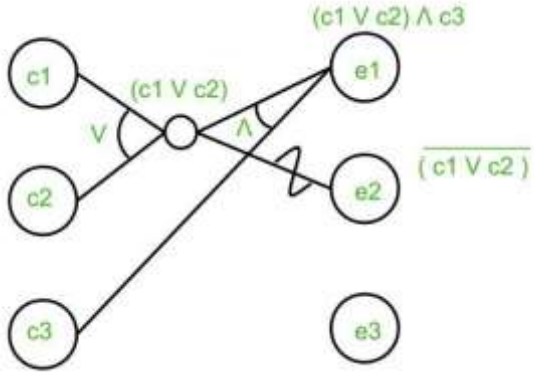
3. Create nodes for all the causes and effects:

4. Use AND, NOT, OR and Identity functions to establish links between causes and effects:

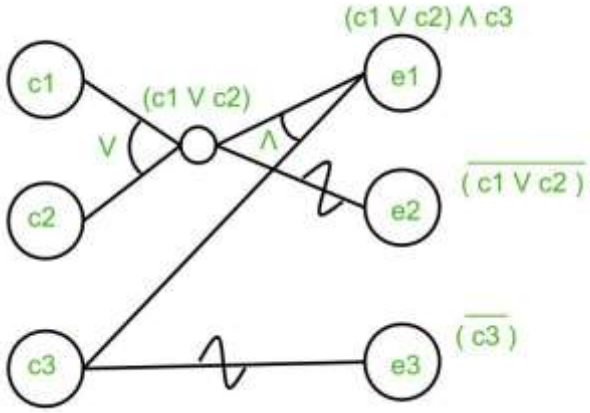
a. e1 is obtained from c1, c2 and c3 in the following manner:



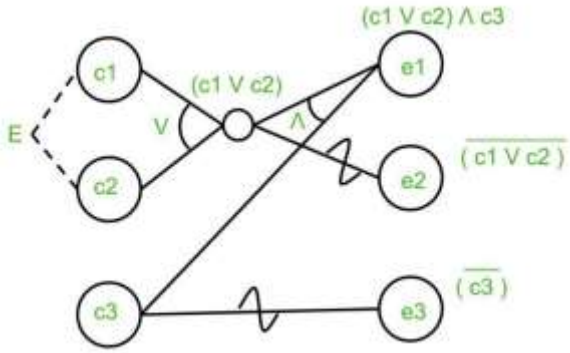
b. e2 is obtained from c1 and c2 in the following manne



c. e3 is obtained from c3 in the following manner:



d. Add constraints to the graph, if any:



Que 2) How many steps are involved in creating cause-effect graph?

Answer) There are total of 5 steps involved in creating cause-effect graph:

- **Step 1:** Identify and Define the Effect
- **Step 2:** Fill in the Effect Box and Draw the Spine
- **Step 3:** Identify the main causes contributing to the effect being studied.
- **Step 4:** For each major branch, identify other specific factors which may be the causes of the EFFECT.
- **Step 5:** Categorize relative causes and provide detailed levels of causes.

Que 3) Which testing technique can be converted into limited entry decision table?

Answer)

Equivalence Partitioning is a testing technique that can be converted into a limited entry decision table.

Equivalence Partitioning involves dividing the input data into a set of partitions or classes based on the assumption that all values in each partition will behave in the same way. This technique is used to reduce the number of test cases by selecting representative values from each partition for testing.

A limited entry decision table is a tabular representation of the conditions and actions associated with a decision. The table has a limited number of entries, and each entry represents a combination of input conditions and corresponding actions.

Equivalence Partitioning can be used to identify the input conditions that need to be considered in a limited entry decision table. The partitions can be used to define the conditions, and the representative values selected from each partition can be used to create the entries in the table.

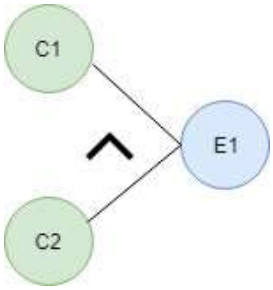
By using Equivalence Partitioning to identify the input conditions and selecting representative values for each partition, the number of test cases required for testing can be reduced, while still ensuring that all relevant combinations of input conditions are considered.

Que 4) Specify the basic notations used in cause-effect graph.

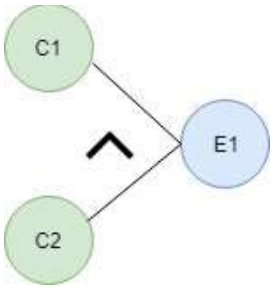
Answer)

Basic notations used in cause-effect graph are:

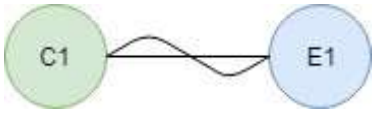
- **AND -** E1 is an effect and C1 and C2 are the causes. If both C1 and C2 are true, then effect E1 will be true.



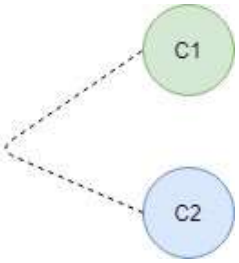
- **OR -** If any cause from C1 and C2 is true, then effect E1 will be true.



- **NOT -** If cause C1 is false, then effect E1 will be true.



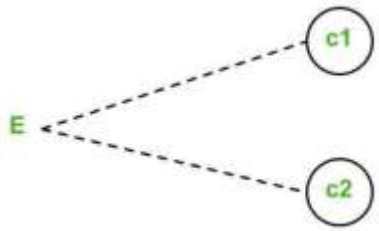
- **Mutually Exclusive -** When only one cause is true.



Que 5) How to represent exclusive or mutual constraints in cause-effect graph?

Answer)

Exclusive constraint or **E-constraint:** This constraint exists between causes. It states that either c1 or c2 can be 1, i.e., c1 and c2 cannot be 1 simultaneously. It is represented as:



Que 6) Give suitable example for cause-effect relationship graph in testing perspective.

Answer)

Cause-effect graph comes under the black box testing technique which underlines the relationship between a given result and all the factors affecting the result. It is used to write dynamic test cases. The dynamic test cases are used when code works dynamically based on user input. For example, while using email account, on entering valid email, the system accepts it but, when you enter invalid email, it throws an error message. In this technique, the input conditions are assigned with causes and the result of these input conditions with effects.

The main advantage of cause-effect graph testing is, it reduces the time of test execution and cost. This technique aims to reduce the number of test cases but still covers all necessary test cases with maximum coverage to achieve the desired application quality. Cause-Effect graph technique converts the requirements specification into a logical relationship between the input and output conditions by using logical operators like AND, OR and NOT.

Example:

Situation: The character in column 1 should be either A or B and in the column 2 should be a digit. If both columns contain appropriate values, then update is made. If the input of column 1 is incorrect, i.e. neither A nor B, then message X will be displayed. If the input in column 2 is incorrect, i.e. input is not a digit, then message Y will be displayed.

- A file must be updated, if the character in the first column is either "A" or "B" and in the second column it should be a digit.
- If the value in the first column is incorrect (the character is neither A nor B) then massage X will be displayed.

- If the value in the second column is incorrect (the character is not a digit) then massage Y will be displayed.

Column 1	Column 2
Correct value - A or B	Correct value – Any digit
Incorrect value – Any character except A or B	Incorrect value – Any character except digit

Now, we are going to make a Cause-Effect graph for the above situation:

Causes are:

- C1 - Character in column 1 is A
- C2 - Character in column 1 is B
- C3 - Character in column 2 is digit!

Effects are:

- E1 - Update made (C1 OR C2) AND C3
- E2 - Displays Massage X (NOT C1 AND NOT C2)
- E3 - Displays Massage Y (NOT C3)

Where AND, OR, NOT are the logical gates.