



Cause-Effect Graphing

Dr. Sangharatna Godbole
Assistant Professor
Department of CSE
NIT Warangal

Slides Credits: Dr. Durga Prasad Mohapatra, Professor, Department of CSE, NIT Rourkela

Cause-Effect Graphing

- Cause-effect graphing, also known as *dependency modeling*,
 - focuses on modelling dependency relationships amongst
 - program input conditions, known as *causes*, and
 - output conditions, known as *effects*.
- The relationship is expressed visually in terms of a cause-effect graph.
- The graph is a visual representation of a logical relationship amongst inputs and outputs that can be expressed as a Boolean expression.

Cause-Effect Graphing (Contd..)

- The graph allows selection of various combinations of input values as tests.
- The combinatorial explosion in the number of tests is avoided by using certain heuristics during test generation.

Cause-Effect Graphing (Contd..)

- A cause is any condition in the requirements that may effect the program output.
- An effect is the response of the program to some combination of input conditions.
 - For example, it may be
 - An error message displayed on the screen
 - A new window displayed
 - A database updated.

Cause-Effect Graphing (Contd..)

- An effect need not be an “output” visible to the user of the program.
- Instead, it could also be an internal *test point* in the program that can be probed during testing to check if some intermediate result is as expected.
 - For example, the intermediate test point could be at the entrance into a method to indicate that indeed the method has been invoked.

Example

- Consider the requirement “Dispense food only when the DF switch is ON”
 - Cause is “DF switch is ON”.
 - Effect is “Dispense food”.
- This requirement implies a relationship between the “DF switch is ON” and the effect “Dispense food”.
- Other requirements might require additional causes for the occurrence of the “Dispense food” effect.

Cause and Effect Graphs

- Testing would be a lot easier:
 - if we could automatically generate test cases from requirements.
- Work done at IBM:
 - Can requirements specifications be systematically used to design functional test cases?

Cause and Effect Graphs

- Examine the requirements:
 - restate them as logical relation between inputs and outputs.
 - The result is a Boolean graph representing the relationships
 - called a **cause-effect graph**.

Cause and Effect Graphs

- Convert the graph to a decision table:
 - each column of the decision table corresponds to a test case for functional testing.

Steps to create cause-effect graph

- Study the functional requirements.
- Mark and number all causes and effects.
- Numbered causes and effects:
 - become nodes of the graph.

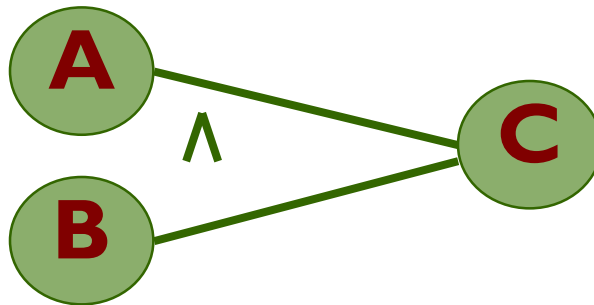
Steps to create cause-effect graph

- Draw causes on the LHS
- Draw effects on the RHS
- Draw logical relationship between causes and effects
 - as edges in the graph.
- Extra nodes can be added
 - to simplify the graph

Drawing Cause-Effect Graphs

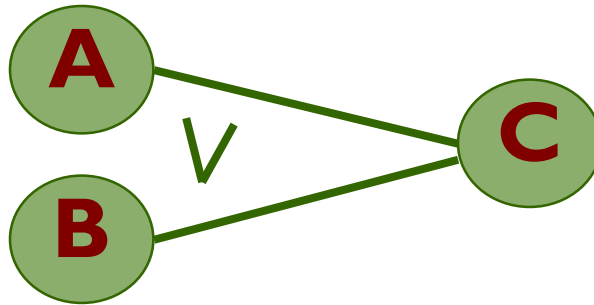


If A then B

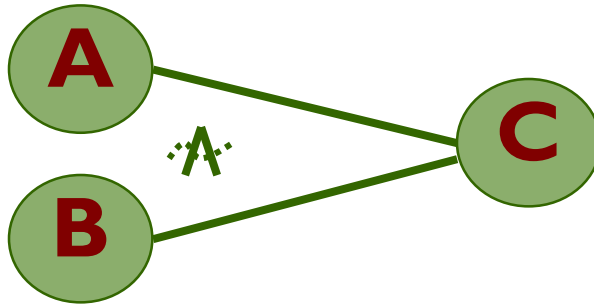


If (A and B) then C

Drawing Cause-Effect Graphs

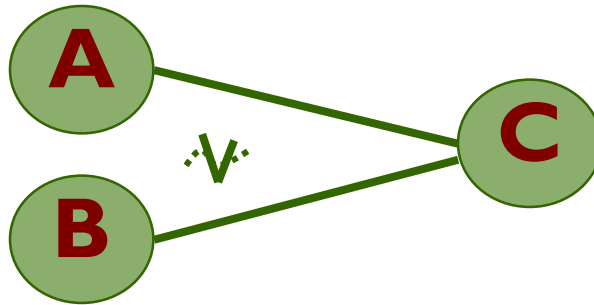


If (A or B) then C

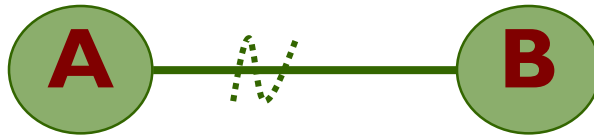


**If (not(A and B)) then
C**

Drawing Cause-Effect Graphs



If (not (A or B)) then C



If (not A) then B

Cause effect graph- Example

- A water level monitoring system
 - used by an agency involved in flood control.
 - **Input:** level(a,b)
 - a is the height of water in dam in meters
 - b is the rainfall in the last 24 hours in cms

Cause effect graph- Example

- Processing
 - The function calculates whether the level is safe, too high, or too low.
- Output
 - message on screen
 - level=safe
 - level=high
 - invalid syntax

Cause effect graph- Example

- We can separate the requirements into 5 clauses:

- 1
 - first five letters of the command is “level”
- 2
 - command contains exactly two parameters
 - separated by comma and enclosed in parentheses

Cause effect graph- Example

3

- Parameters A and B are real numbers:
 - such that the water level is calculated to be low

4

- or safe.

5

- The parameters A and B are real numbers:
 - such that the water level is calculated to be high.

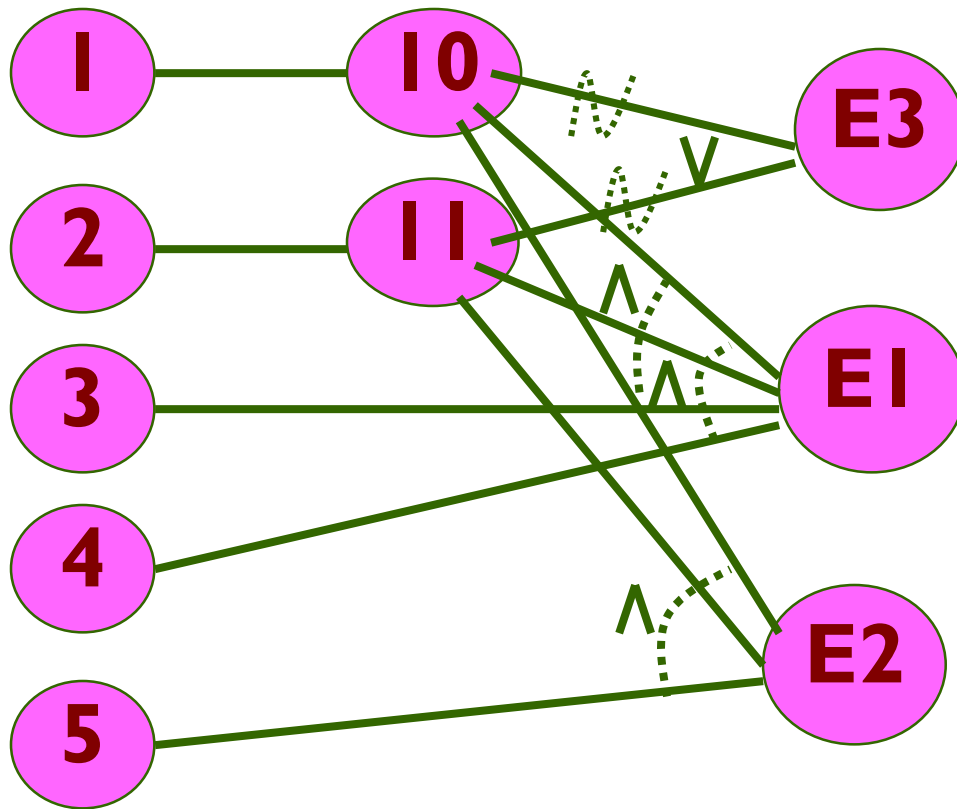
Cause effect graph- Example

- 10 ◦ Command is syntactically valid
- 11 ◦ Operands are syntactically valid.

Cause effect graph- Example

- Three effects
 - level = safe E1
 - level = high E2
 - invalid syntax E3

Cause effect graph- Example



Cause effect graph- Decision table

	Test 1	Test 2	Test 3	Test 4	Test 5	
Cause 1	I	I	I	S	I	I = Invoked x = don't care s = suppressed
Cause 2	I	I	I	X	S	
Cause 3	I	S	S	X	X	
Cause 4	S	I	S	X	X	
Cause 5	S	S	I	X	X	
Effect 1	P	P	A	A	A	P = present A = absent
Effect 2	A	A	P	A	A	
Effect 3	A	A	A	P	P	

Cause effect graph- Example

- Put a row in the decision table for each cause or effect:
 - in the example, there are five rows for causes and three for effects.

Cause effect graph- Example

- The columns of the decision table correspond to test cases.
- Define the columns by examining each effect:
 - list each combination of causes that can lead to that effect.

Cause effect graph- Example

- We can determine the number of columns of the decision table
 - by examining the lines flowing into the effect nodes of the graph.

Cause effect graph- Example

- Theoretically we could have generated $2^5=32$ test cases.
 - Using cause effect graphing technique reduces that number to 5.

Cause effect graph

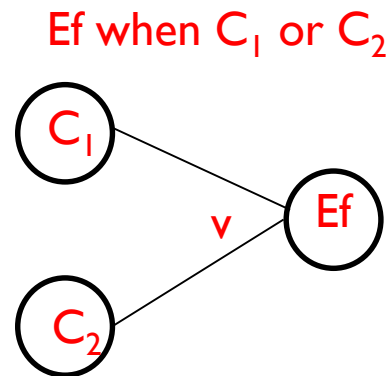
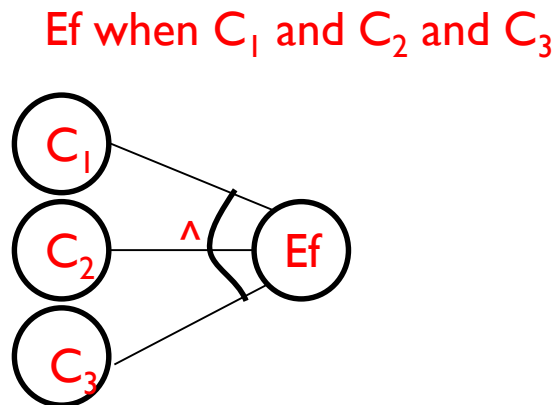
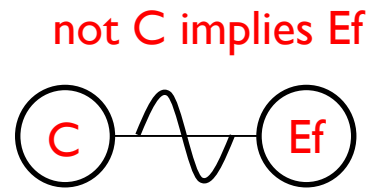
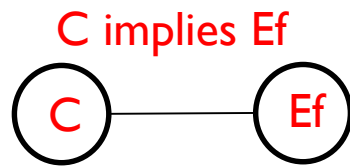
- Not practical for systems which:
 - include timing aspects
 - feedback from processes is used for some other processes.

Procedure used for the generation of tests

- Identify causes and effects by reading the requirements. Each cause and effect is assigned a unique identifier. Note that an effect can also be a cause for some other effect.
- Express the relationship between causes and effects using a cause-effect graph.
- Transform the cause-effect graph into a limited entry decision table, hereafter referred to as decision table.
- Generate tests from the decision table.

Basic elements of a cause-effect graph

- implication
- not (\sim)
- and (\wedge)
- or (\vee)



- C, C_1, C_2, C_3 denote causes.
- Ef denotes an effect.

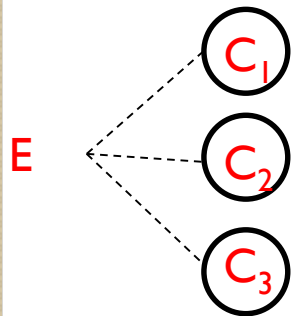
Semantics of basic elements

- C implies Ef : $\text{if}(C) \text{ then } Ef;$
- not C implies Ef : $\text{if}(\neg C) \text{ then } Ef;$
- Ef when C_1 and C_2 and C_3 : $\text{if}(C_1 \&\&C_2 \&\&C_3) \text{ then } Ef;$
- Ef when C_1 or C_2 : $\text{if}(C_1 || C_2) \text{ then } Ef;$

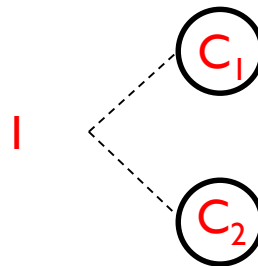
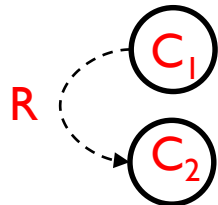
Constraints amongst causes (E,I,O,R)

- Constraints show the relationship between the causes.
- Exclusive (E)
- Inclusive (I)
- Requires (R)
- One and only one (O)

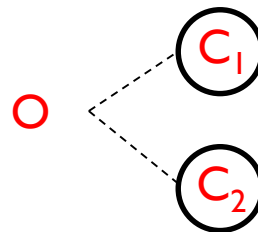
Exclusive: either C_1 or C_2 or C_3 Inclusive: at least C_1 or C_2



C_1 requires C_2



One and only one, of C_1 and C_2



Constraints amongst causes (E,I,O,R)

- Exclusive (E) constraint between three causes C_1 , C_2 and C_3 implies that exactly one of C_1 , C_2 , C_3 can be true.
- Inclusive (I) constraint between two causes C_1 and C_2 implies that at least one of the two must be present.
- Requires (R) constraint between C_1 and C_2 implies that C_1 requires C_2 .
- One and only one (O) constraint models the condition that one, and only one, of C_1 and C_2 must hold.

Possible values of causes constrained by E, I, R, O

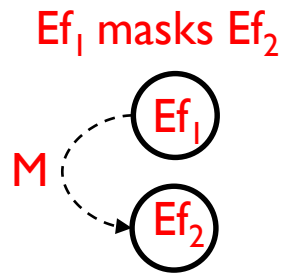
- A 0 or 1 under a cause implies that the corresponding condition is, respectively, false and true.
- The arity of all constraints, except R, is greater than 1, i.e., all except the R constraint can be applied to two or more causes; the R constraint is applied to two causes.
- A condition that is false (true) is said to be in the “0-state” (1 state).
- Similarly, an effect can be “present” (1 state) or “absent” (0 state).

Possible values of causes constrained by E, I, R, O

Constraint	Arity	Possible values		
		C1	C2	C3
$E(C_1, C_2, C_3)$	$n \geq 2$	0	0	0
		1	0	0
		0	1	0
		0	0	1
$I(C_1, C_2)$	$n \geq 2$	1	0	-
		0	1	-
		1	1	-
$R(C_1, C_2)$	$n = 2$	1	1	-
		0	0	-
		0	1	-
$O(C_1, C_2, C_3)$	$n \geq 2$	1	0	0
		0	1	0
		0	0	1

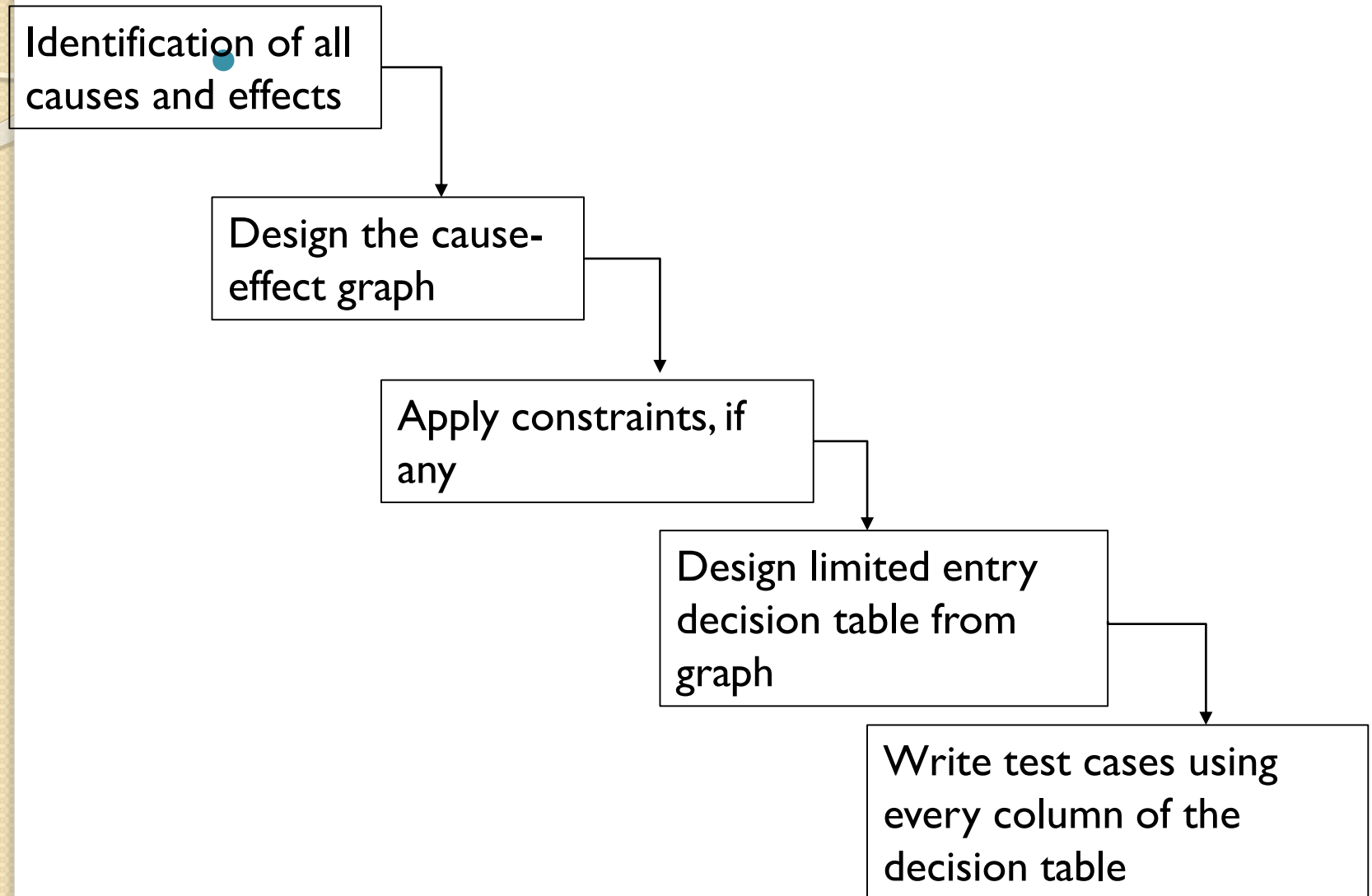
Constraint amongst effects

- Masking (M)



- Masking (M) constraint between two effects Ef_1 and Ef_2 implies that if Ef_1 is present, then Ef_2 is forced to be absent.

Steps for generating test cases using Cause-Effect Graph



Creating Cause-Effect Graph

- The process of creating a cause-effect graph consists of two major steps.
- The causes and effects are identified by a careful examination of the requirements.
 - This process also exposes the relationships amongst various causes and effects as well as constraints amongst the causes and effects.
 - Each cause and effect is assigned a unique identifier for ease of reference in the cause-effect graph.

Creating Cause-Effect Graph

- The cause-effect graph is constructed to
 - express the relationships extracted from the requirements.
- When the number of causes and effects is large, say over 100 causes and 45 effects,
 - it is appropriate to use an incremental approach.

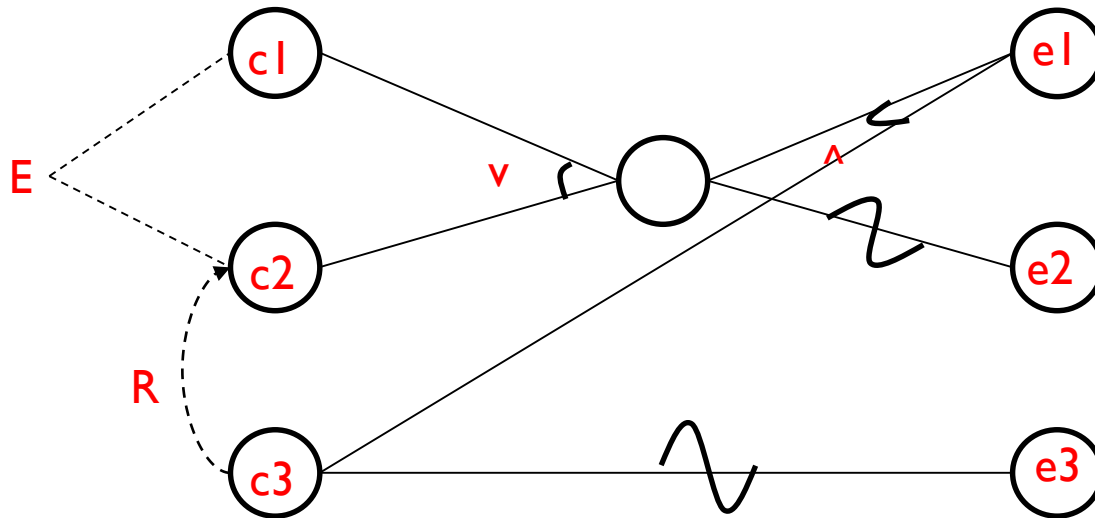
Another example

- Consider the example of keeping the record of marital status and number of children of a citizen.
- The value of marital status must be 'U' or 'M'.
- The value of the number of children must be digit or null in case a citizen is unmarried.
- If the information entered by the user is correct then an update is made.
- If the value of marital status of the citizen is incorrect, then the error message 1 is issued.
- Similarly, if the value of the number of children is incorrect, then the error message 2 is issued.

Answer

- Causes are
 - c1: marital status is U
 - c2: marital status is M
 - c3: number of children is a digit
- Effects are
 - e1: updation made
 - e2: error message 1 is issued
 - e3: error message 2 is issued

Answer



- There are two constraints
 - Exclusive (between c1 and c2) and
 - Requires (between c3 and c2)
- Causes c1 and c2 cannot occur simultaneously.
- For cause c3 to be true, cause c2 has to be true.

Decision Table from cause-effect graph

- Each column of the decision table represents a combination of input values, and hence a test.
- There is one row for each condition and effect.
- Thus the table decision table can be viewed as an $N \times M$ matrix with
 - N being the sum of the number of conditions and effects and
 - M the number of tests.
- Each entry in the decision table is a 0 or 1
 - depending on whether or not the corresponding condition is false or true, respectively.
- For a row corresponding to an effect, an entry is 0 or 1
 - if the effect is not present or present, respectively.

Test generation from a decision table

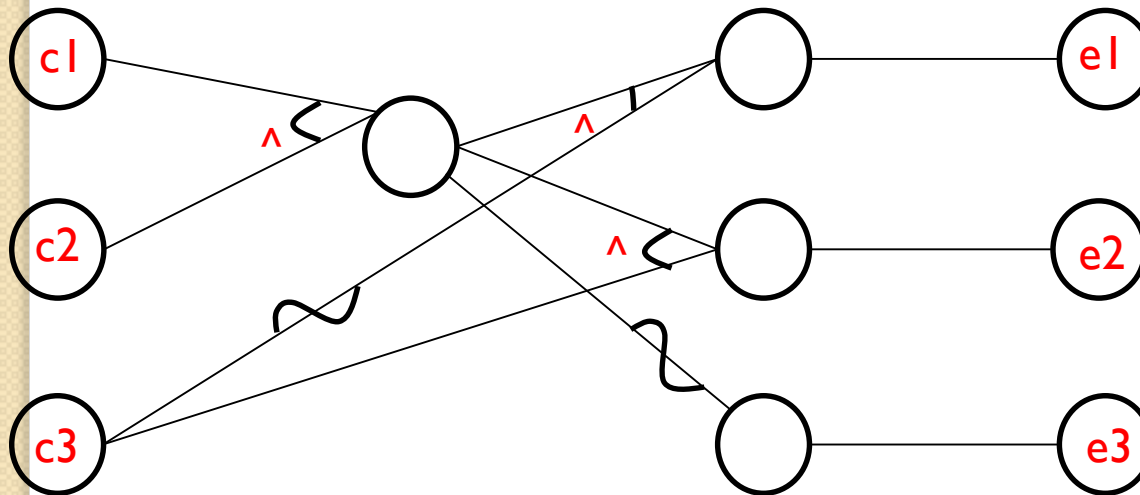
- Test generation from a decision table is relatively forward.
- Each column in the decision table generates at least one test input.
- Note that each combination might be able to generate more than one test when a condition in the cause-effect graph can be satisfied in more than one way.
- For example, consider the following cause:
- $C: x < 99$
- The condition above can be satisfied by many values such as $x=1$ and $x=49$.
- Also, C can be made false by many values of x such as $x=100$ and $x=999$.
- Thus, one might have a choice of values of input variables while generating tests using columns from a decision table

Example

- A tourist of age greater than 21 years and having a clean driving record is supplied a rental car.
- A premium amount is also charged if the tourist is on business,
- Otherwise, it is not charged.
- If the tourist is less than 21 year old, or does not have a clean driving record,
 - The system will display the following message: “Car cannot be supplied”.

Answer

- Causes are
 - c1: Age is over 21
 - c2: Driving record is clean
 - c3: Tourist is on business
- Effects are
 - e1: Supply a rental car without premium charge
 - e2: Supply a rental car with premium charge
 - e3: Car cannot be supplied



Decision Table and Test Cases

	1	2	3	4
c1: Over 21?	F	T	T	T
c2: Driving record clean?	-	F	T	T
c3: On business?	-	-	F	T
e1: Supply a rental car without premium charge			X	
e2: Supply a rental car with premium charge				X
e3: Car cannot be supplied	X	X		

Test Case	Age	Driving_record_clean	On_business	Expected Output
1	20	Yes	Yes	Car cannot be supplied
2	26	No	Yes	Car cannot be supplied
3	62	Yes	No	Supply a rental car without premium charge
4	62	Yes	Yes	Supply a rental car with premium charge

Example 2: Triangle Classification Problem

- Consider a program for classification of a triangle.
- Its input is a triple of positive integers (say a , b and c) and the input values are greater than zero and less than or equal to 100.
- The triangle is classified according to the following rules:
 - Right angled triangle: $c^2 = a^2 + b^2$ or $a^2 = b^2 + c^2$ or $b^2 = c^2 + a^2$
 - Obtuse angled triangle: $c^2 > a^2 + b^2$ or $a^2 > b^2 + c^2$ or $b^2 > c^2 + a^2$
 - Acute angled triangle: $c^2 < a^2 + b^2$ or $a^2 < b^2 + c^2$ or $b^2 < c^2 + a^2$
 - The program output may have one of the following words: [Acute angled triangle, Obtuse angled triangle, Right angled triangle, Invalid triangle, Input values are out of range]

Answer

- Causes are:
 - c1: side “a” is less than the sum of sides “b” and “c”.
 - c2: side “b” is less than the sum of sides “a” and “c”.
 - c3: side “c” is less than the sum of sides “a” and “b”.
 - c4: square of side “a” is equal to the sum of squares of sides “b” and “c”.
 - c5: square of side “a” is greater than the sum of squares of sides “b” and “c”.
 - c6: square of side “a” is less than the the sum of squares of sides “b” and “c”.
- Effects are:
 - e1: Invalid triangle
 - e2: Right angle triangle
 - e3: Obtuse angled triangle
 - e4: Acute angled triangle
 - e5: Impossible stage



Decision Table

	1	2	3	4	5	6	7	8	9	10	11
c1: $a < b + c$	F	T	T	T	T	T	T	T	T	T	T
c2: $b < a + c$	-	F	T	T	T	T	T	T	T	T	T
c3: $c < a + b$	-	-	F	T	T	T	T	T	T	T	T
c4: $a^2 = b^2 + c^2$	-	-	-	T	T	T	T	F	F	F	F
c5: $a^2 > b^2 + c^2$	-	-	-	T	T	F	F	T	T	F	F
c6: $a^2 < b^2 + c^2$	-	-	-	T	F	T	F	T	F	T	F
e1: Invalid triangle	X	X	X								
e2: Right angle triangle							X				
e3: Obtuse angled triangle									X		
e4: Acute angled triangle										X	
e5: Impossible				X	X	X		X			X



Thank You