Cpt S 422: Software Engineering Principles II

Black-box testing – Part 3

Dr. Venera Arnaoudova

Black-box testing methods

- ✓ Equivalence Class Partitioning
- ✓ Boundary-Value Analysis
- ✓ Category-Partition
- ✓ Decision tables
- ❑ Cause-Effect Graphs
- ❑ Logic Functions

Cause-Effect Graphs Testing

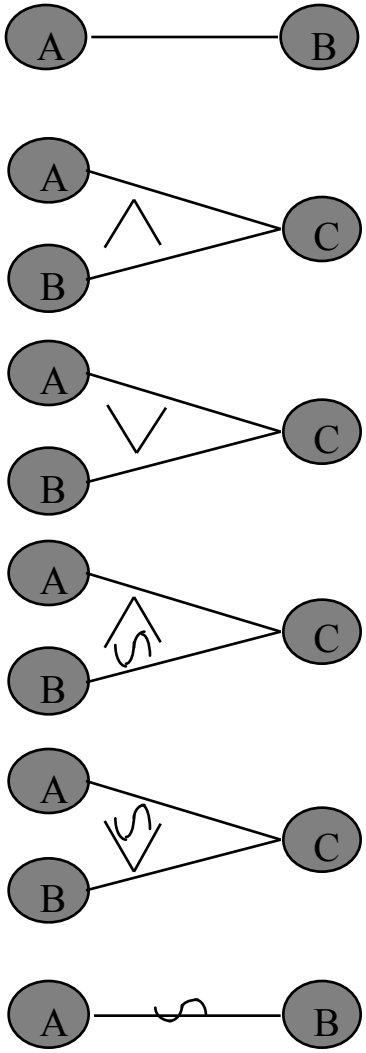
Definition

- Graphical technique that helps derive decision tables
- Aim at supporting interaction with domain experts and the reverse engineering of specifications, for the purpose of testing.
- Identify causes (conditions on inputs, stimuli) and effects (outputs, changes in system state)
- Causes have to be stated in such a way to be either true or false (Boolean expression)
- Specifies explicitly (environmental, external) constraints on causes and effects
- Help select more “significant” subset of input-output combinations and build smaller decision tables

Structure of Cause-Effect Graphs

- A node is drawn for each cause and effect
- Nodes placed on opposite side of a sheet
- A line from a cause to an effect indicates that the cause is a necessary condition for the effect
- If a single effect has two or more causes, the logical relationship of the causes is annotated by symbols for logical *and* (^) and logical *or* ($\vee$) placed between lines
- A cause whose negation is necessary is shown by a logical *not* (~)
- A single cause may be necessary for many effects; a single effect may have many necessary causes.
- Intermediate nodes may be used to simplify the graph and its construction

Notation



If A then B

AND: If (A and B) then C

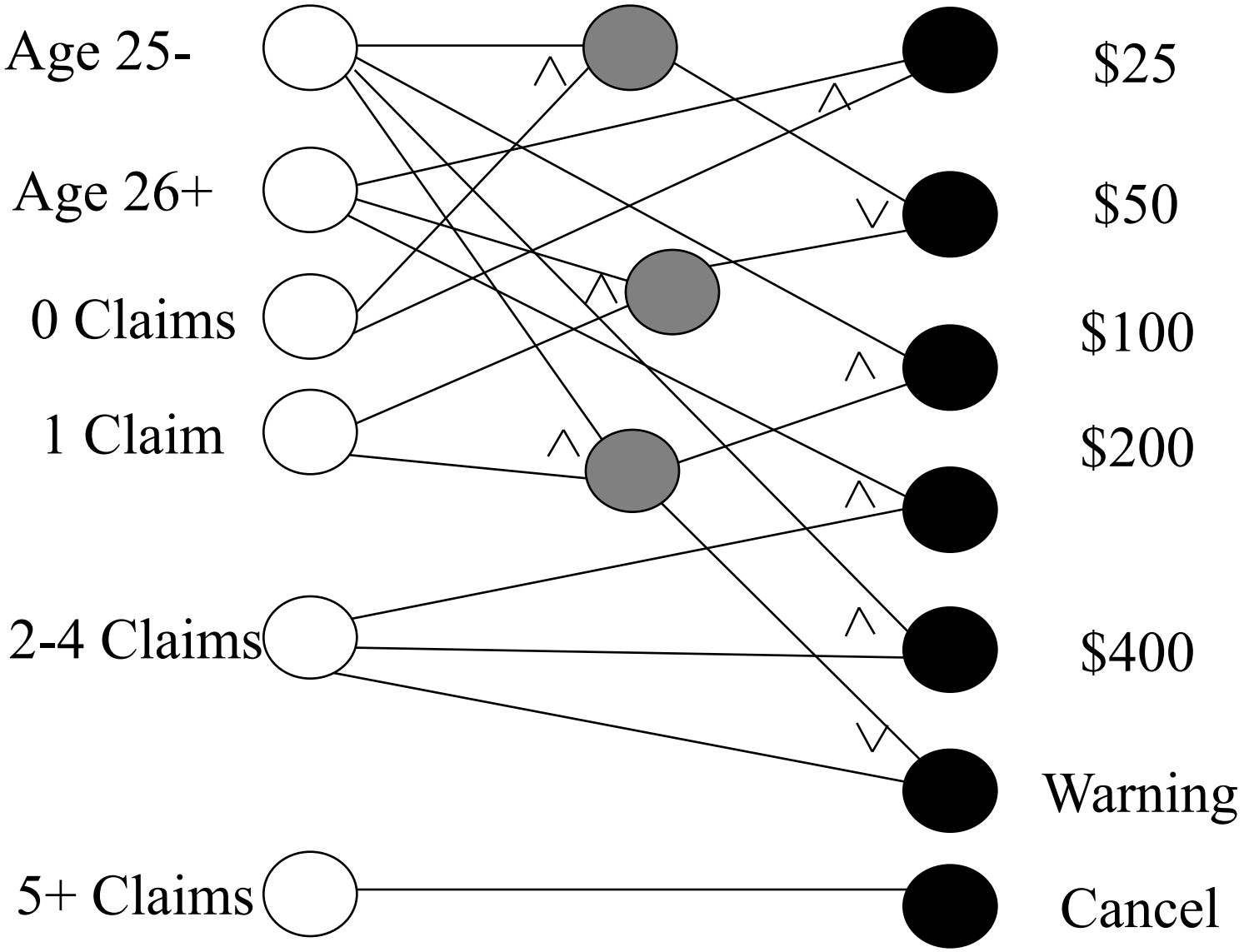
OR: If (A or B) then C

NAND: If not (A and B) then C

NOR: If (neither A nor B) then C

NOT: If (not A) then B

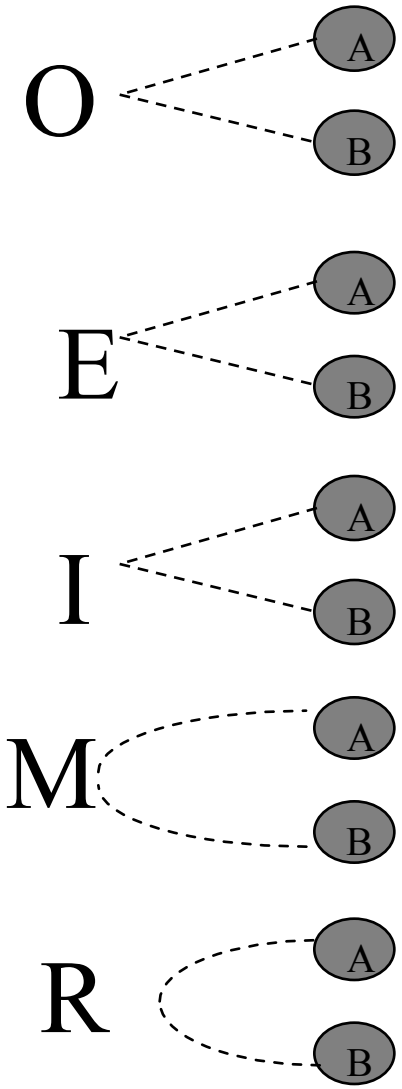
Insurance Renewal Example



Another Table Example Insurance Renewal

	Condition Section		Action Section		
Variant	**Claims**	**Age**	**Premium Increase $**	**Send Warning**	**Cancel**
1	0	25-	50	No	No
2	0	26+	25	No	No
3	1	25-	100	Yes	No
4	1	26+	50	No	No
5	2 to 4	25-	400	Yes	No
6	2 to 4	26+	200	Yes	No
7	5+	Any	0	No	Yes

Additional Constraints



EXACTLY ONE of A and B must be true

AT MOST ONE of A and B may be true

AT LEAST ONE of A and B must be true

A MASKS B, i.e., A => NOT B

A REQUIRES B, i.e., A => B

Another Example

- Input: The syntax of the function is LEVEL(A,B) where A is the height in meters of the water behind the dam and B is the number of centimeters of rain in the last 24-hour period.

- Processing: The function calculates whether the water level is (1) within a normal range, (2) too high, (3) too low

- Outputs: one of the following messages
  - LEVEL = SAFE (for normal and low)
  - LEVEL = HIGH
  - INVALID SYNTAX

Identifying Causes

1. Five first characters of command: LEVEL
2. The two parameters separated by a comma and enclosed in parentheses
3. The parameters A and B are real numbers such that the water level is calculated to be LOW
4. The parameter A and B are real numbers such that the water level is calculated to be NORMAL
5. The parameter A and B are real numbers such that the water level is calculated to be HIGH

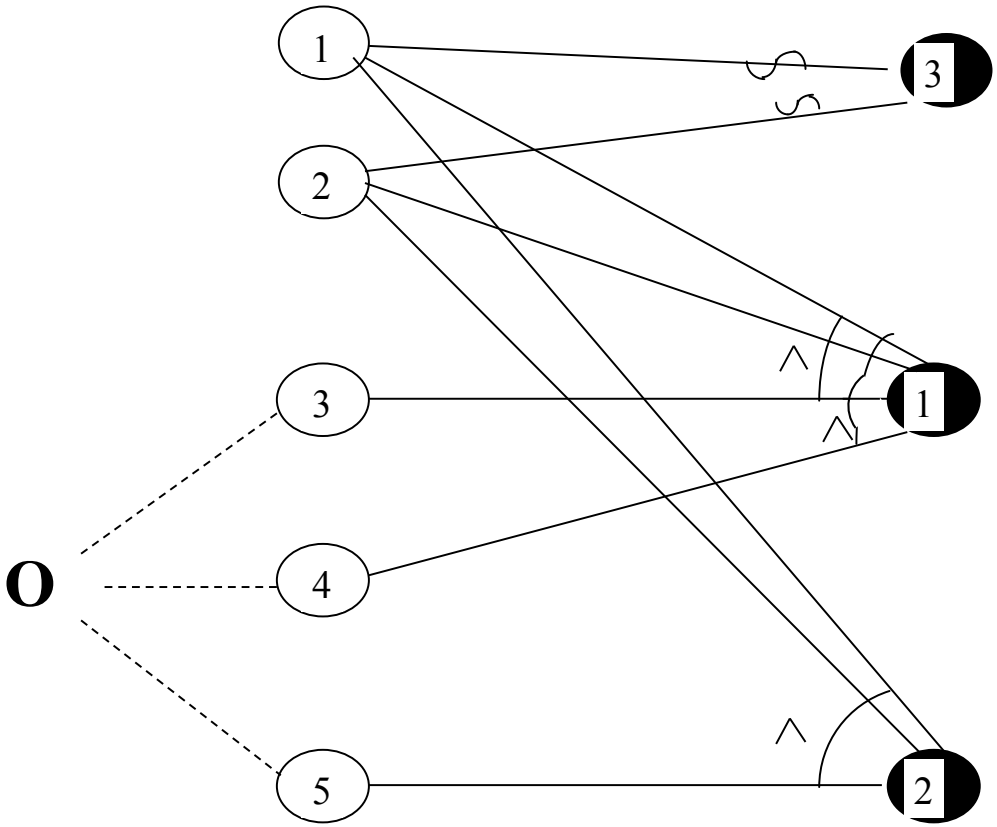
Identifying Effects

1. “LEVEL = SAFE”
2. “LEVEL = HIGH”
3. “INVALID SYNTAX”

Cause-Effect Graph for LEVEL

Causes:

1. Five first characters of command: LEVEL
2. The two parameters separated by a comma and enclosed in parentheses
3. The parameter A and B are real numbers such that the water level is calculated to be LOW
4. The parameter A and B are real numbers such that the water level is calculated to be NORMAL
5. The parameter A and B are real numbers such that the water level is calculated to be HIGH



Effects:

1. "LEVEL = SAFE"
2. "LEVEL = HIGH"
3. "INVALID SYNTAX"

Deriving a Decision Table

- A row for each cause or effect
- The columns correspond to test cases (variants)
- We define columns by examining each effect and listing all combinations (conjunctions) of causes that can lead to that effect
- E.g., two separate lines flow into effect E3, each corresponding to a test case, four lines flow into E1 but correspond to only two combinations

LEVEL Decision Table

Decision Table for Cause-and-effect Graph

	Test 1	Test 2	Test 3	Test 4	Test 5
Cause 1	T	T	T	F	T
Cause 2	T	T	T	-	F
Cause 3	T	F	F	-	-
Cause 4	F	T	F	-	-
Cause 5	F	F	T	-	-
Effect 1	P	P	A	A	A
Effect 2	A	A	P	A	A
Effect 3	A	A	A	P	P

Discussion

- The Cause-Effect graph can be used to generate all *possible* combinations of causes and checking whether the effect corresponds to the specification
- It provides a test oracle and specifies constraints on outputs (effects), helping detecting wrong system states and action combinations
- If the graph is too large, for each admissible combination of effects, find some combinations of causes that cause that combination of effects by tracing back through the graph
- Because of additional constraints on graph, can be more restrictive than straight decision tables