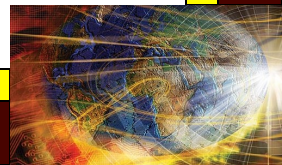


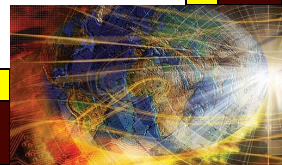
Chapter 7

Decision Table-Based Testing



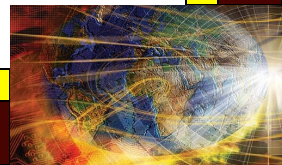
Outline

- Decision table vocabulary
 - Limited Entry Decision Tables (LEDT)
 - Extended Entry Decision Tables (EEDT)
 - Mixed Entry Decision Tables (MEDT)
- Techniques
 - Redundant LEDTs
 - Inconsistent LEDTs
- Examples
- Guidelines
- Cause and Effect Graphs



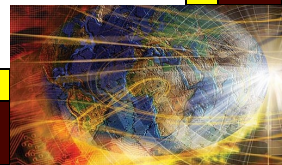
Decision Table Based Testing

- Originally known as Cause and Effect Graphing
 - Done with a graphical technique that expressed AND-OR-NOT logic.
 - Causes and Effects were graphed like circuit components
 - Inputs to a circuit “caused” outputs (effects)
- Equivalent to forming a decision table in which:
 - inputs are conditions
 - outputs are actions
- Test every (possible) rule in the decision table.
- Recommended for logically complex situations.
- Excellent example of Model-Based Testing (MBT)



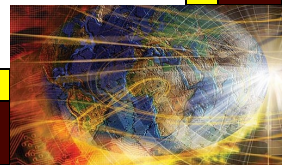
Decision Tables

- Represent complex conditional behavior.
- Support extensive analysis
 - Consistency
 - Completeness
 - Redundancy
 - Algebraic simplification
- Executable (and compilable)
- Two forms: Limited and Extended Entry.
- “Don't Care” condition entries require special attention.
- Dependencies usually yield impossible situations



Content of a Decision Table

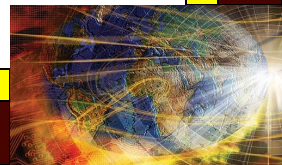
- Conditions
 - binary in a Limited Entry Decision Table
 - finite set in an Extended Entry Decision Table
 - condition stub
 - condition entries
- Actions
 - also binary, either do or skip
 - the “impossible” action
- Rules
 - a rule consists of condition entries and action entries
 - a complete, non-redundant LEDT with n conditions has 2^n rules
 - logically impossible combinations of conditions are “impossible rules”, denoted by an entry in the impossible action



Example

<i>Stub</i>	<i>Rule 1</i>	<i>Rule 2</i>	<i>Rule 3</i>	<i>Rule 4</i>	<i>Rule 5</i>	<i>Rule 6</i>	<i>Rule 7</i>	<i>Rule 8</i>
c1	T	T	T	T	F	F	F	F
c2	T	T	F	F	T	T	F	F
c3	T	F	T	F	T	F	T	F
a1	X	X			X			
a2	X					X		
a3		X						
a4			X	X	X		X	X

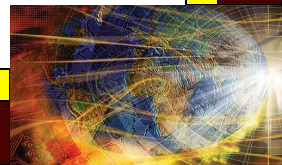
Condition c3 has no effect on the actions of Rules 3 and 4. Similarly for Rules 7 and 8. They can be algebraically combined.



Example (continued)

<i>Stub</i>	<i>Rule 1</i>	<i>Rule 2</i>	<i>Rules 3, 4</i>	<i>Rule 5</i>	<i>Rule 6</i>	<i>Rules 7, 8</i>
c1	T	T	T	F	F	F
c2	T	T	F	T	T	F
c3	T	F	—	T	F	T
a1	X	X		X		—
a2	X				X	
a3		X				
a4			X	X		X

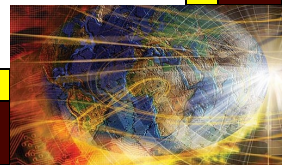
- The condition entries in rules 3 and 4, and rules 7 and 8 have the same actions. The “—” means ...
 - “Don’t Care” (as in circuit analysis),
 - Irrelevant, or
 - not applicable, n/a



Example (continued)

<i>Stub</i>	<i>Rule 1</i>	<i>Rule 2</i>	<i>Rules 3, 4, 7, 8</i>	<i>Rule 5</i>	<i>Rule 6</i>
c1	T	T	—	F	F
c2	T	T	F	T	T
c3	T	F	—	T	F
a1	X	X		X	
a2	X				X
a3		X			
a4			X	X	

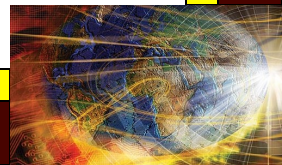
- One more algebraic simplification



Example (continued)

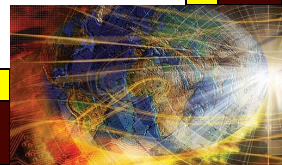
<i>Stub</i>	<i>Rule 1</i>	<i>Rule 2</i>	<i>Rules 3, 4, 7, 8</i>	<i>Rule 5</i>	<i>Rule 6</i>
c1	T	T	—	F	F
c2	T	T	F	T	T
c3	T	F	—	T	F
a1	X	X		X	
a2	X				X
a3		X			
a4			X	X	
count	1	1	4	1	1

- Rule counting
 - a rule with no don't care entries counts as 1
 - each don't care entry in a rule doubles the rule count
 - for a table with n limited entry conditions, the sum of the rule counts should be 2^n .



Problematic Decision Tables

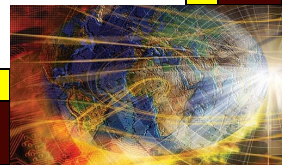
- For LEDTs, simple rule counting helps identify decision tables that are ...
 - incomplete (rule count $< 2^n$) ,
 - redundant (rule count $> 2^n$) , or
 - inconsistent
 - (rule count $> 2^n$) AND
 - at least two rules have identical condition entries but different action entries.
- Redundancy and inconsistency are more likely with algebraically simplified tables that have been “maintained”.



A Redundant DT

<i>conditions</i>	1 – 4	5	6	7	8	9
c1	T	F	F	F	F	T
c2	—	T	T	F	F	F
c3	—	T	F	T	F	F
a1	X	X	X	—	—	X
a2	—	X	X	X	—	—
a3	X	—	X	X	X	X

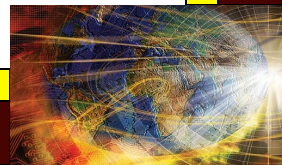
- Rule 9 is redundant with Rules 1 – 4 (technically, with what was rule 4)
- But the action entries are identical (No harm, no foul?)



An Inconsistent DT

<i>conditions</i>	<i>1 – 4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
c1	T	F	F	F	F	T
c2	—	T	T	F	F	F
c3	—	T	F	T	F	F
a1	X	X	X	—	—	—
a2	—	X	X	X	—	X
a3	X	—	X	X	X	—

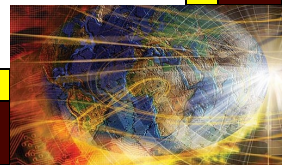
- Rule 9 is inconsistent with Rules 1 – 4 (technically, with what was rule 4)
 - condition portion is identical, BUT
 - action portion is different
- What happens when Rule 4 is executed? Rule 9?



Last Day of Month Decision Table

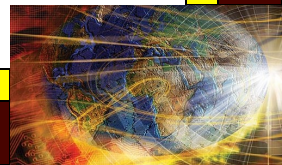
<i>conditions</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c1. month in M1?	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
c2. month in M2?	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
c3. month in M3?	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
c4. leap year?	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
a1. last day = 30								x								
a2. last day = 31											x	x				
a3. last day = 28														x		
a4. last day = 29													x			
a5. impossible	x	x	x	x	x	x			x	x					x	x

- Rule pairs 1 and 2, 3 and 4, 5 and 6, 9 and 10 don't need c4, so they could be combined, BUT
- Impossible because c1, c2, and c3 are mutually exclusive.



Extended Entry Decision Tables

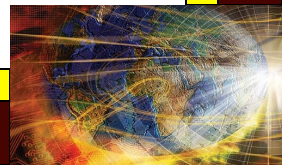
- When conditions are mutually exclusive, exactly one must be true.
- Extended entry decision tables typically (but not necessarily) have mutually exclusive conditions.
- The “extended” part is because a condition stub is an incomplete statement that is completed by the condition entry.
- (See the revised Last Day of Month EEDT)



Revised Last Day of Month DT

c1. month in	M1	M1	—	—	—	—
c2. month in	—	—	M2	M2	—	—
c3. month in	—	—	—	—	M3	M3
c4. leap year?	T	F	T	F	T	F
a1. last day = 30	x	x				
a2. last day = 31			x	x		
a3. last day = 28						x
a4. last day = 29					x	

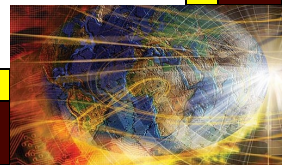
- When conditions are mutually exclusive, exactly one must be true.
- This can be further simplified.



The “Emphatic False”

c1. month in	M1	—	—	—
c2. month in	—	M2	—	—
c3. month in	—	—	M3	M3
c4. leap year?	—	—	T	F
a1. last day = 30	x			
a2. last day = 31		x		
a3. last day = 28				x
a4. last day = 29			x	

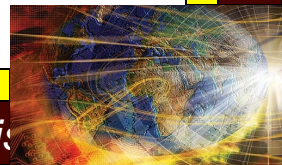
- Maybe “—” should be replaced by “must be False”
- One writer suggested “F!” (before “!” meant “NOT”)
- It is a Don’t Care in c4.
- Technically, this is a Mixed Entry Decision Table, because it has both extended and limited entry conditions.



Triangle Program Decision Table

c1: a, b, c form a triangle?	F	T	T	T	T	T	T	T	T
c2: a = b?	—	T	T	T	T	F	F	F	F
c3: a = c?	—	T	T	F	F	T	T	F	F
c4: b = c?	—	T	F	T	F	T	F	T	F
a1: Not a triangle	X								
a2: Scalene									X
a3: Isosceles					X		X	X	
a4: Equilateral		X							
a5: Impossible			X	X		X			

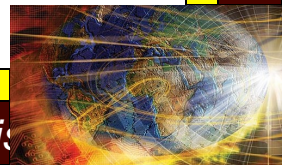
Why are some rules impossible?



Expanding c1...

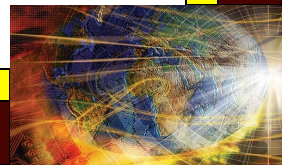
c1: $a < b + c$?	F	T	T	T	T	T	T	T	T	T	T
c2: $b < a + c$?	—	F	T	T	T	T	T	T	T	T	T
c3: $c < a + b$?	—	—	F	T	T	T	T	T	T	T	T
c4: $a = b$?	—	—	—	T	T	T	T	F	F	F	F
c5: $a = c$?	—	—	—	T	T	F	F	T	T	F	F
c6: $b = c$?	—	—	—	T	F	T	F	T	F	T	F
a1: Not a triangle	x	x	x								
a2: Scalene											x
a3: Isosceles						x			x	x	
a4: Equilateral				x							
a5: Impossible					x	x		x			

- Is this a complete decision table?
How many test cases does this imply?



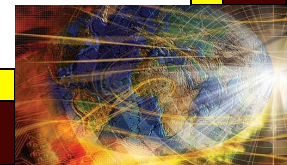
Rule Counting

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: $a = b$?	—	—	—	T	T	T	T	F	F	F	F
c5: $a = c$?	—	—	—	T	T	F	F	T	T	F	F
c6: $b = c$?	—	—	—	T	F	T	F	T	F	T	F
Rule count	32	16	8	1	1	1	1	1	1	1	1
a1: Not a triangle	x	x	x								
a2: Scalene											x
a3: Isosceles						x			x	x	
a4: Equilateral				x							
a5: Impossible					x	x		x			



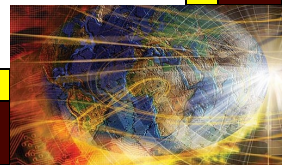
Corresponding Test Cases

<i>Case ID</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>Expected Output</i>
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



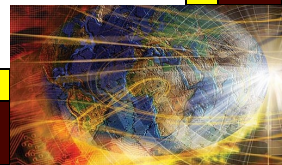
NextDate Decision Table (first half)

	1	2	3	4	5	6	7	8	9	10
c1: month in	M1	M1	M1	M1	M1	M2	M2	M2	M2	M2
c2: day in	D1	D2	D3	D4	D5	D1	D2	D3	D4	D5
c3: year in	—	—	—	—	—	—	—	—	—	—
a1: impossible					X					
a2: increment day	X	X	X			X	X	X	X	
a3: reset day				X						X
a4: increment month				X						X
a5: reset month										
a6: increment year										



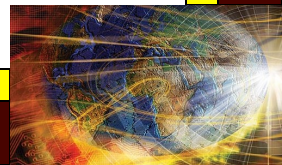
NextDate Decision Table (first half reduced)

	1–3	4	5	6–9	10
c1: month in	M1	M1	M1	M2	M2
c2: day in	D1, D2, D3	D4	D5	D1, D2, D3, D4	D5
c3: year in	—	—	—	—	—
a1: impossible			X		
a2: increment day	X			X	
a3: reset day		X			X
a4: increment month		X			X
a5: reset month					
a6: increment year					



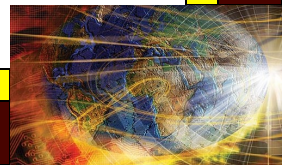
NextDate Decision Table (second half)

	11	12	13	14	15	16	17	18	19	20	21	22
c1: month in	M3	M3	M3	M3	M3	M4	M4	M4	M4	M4	M4	M4
c2: day in	D1	D2	D3	D4	D5	D1	D2	D2	D3	D3	D4	D5
c3: year in	—	—	—	—	—	—	Y1	Y2	Y1	Y2	—	—
a1: impossible										X	X	X
a2: increment day	X	X	X	X		X	X					
a3: reset day					X			X	X			
a4: increment month								X	X			
a5: reset month					X							
a6: increment year					X							



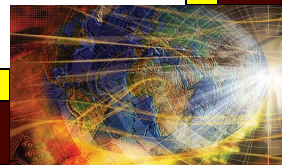
NextDate Decision Table (second half reduced)

	11–14	15	16	17	18	19	20	21, 22
c1: month in	M3	M3	M4	M4	M4	M4	M4	M4
c2: day in	D1, D2, D3, D4	D5	D1	D2	D2	D3	D3	D4, D5
c3: year in	—	—	—	Y1	Y2	Y1	Y2	—
a1: impossible							X	X
a2: increment day	X			X				
a3: reset day		X	X		X	X		
a4: increment month					X	X		
a5: reset month			X					
a6: increment year			X					

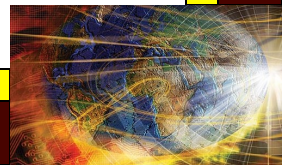
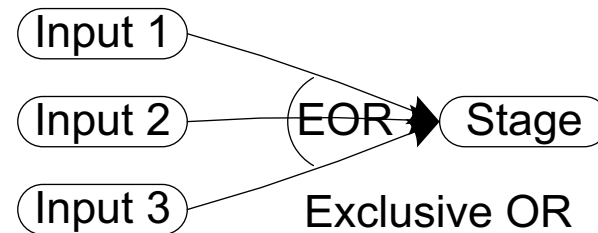
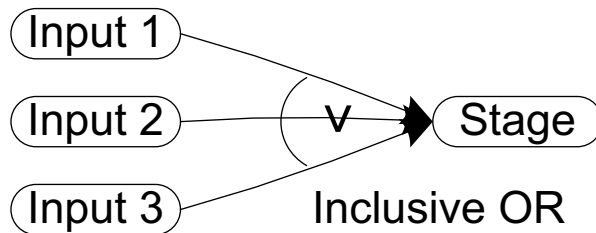
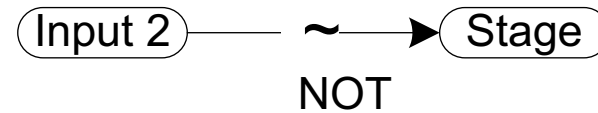
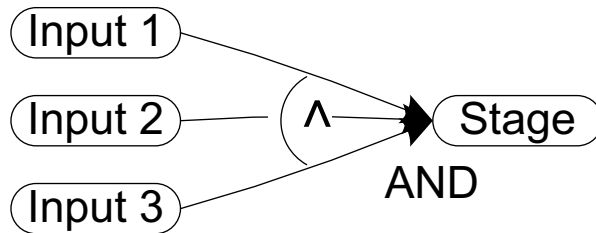


NextDate Test Cases

Test Case	<i>Rule(s)</i>	<i>Month</i>	<i>Day</i>	<i>Year</i>	<i>Expected Output</i>
1	1– 3	4	15	2001	4/16/2001
2	4	4	30	2001	5/1/2001
3	5	4	31	2001	Invalid Input Date
4	6–9	1	15	2001	1/16/2001
5	10	1	31	2001	2/1/2001
6	11–14	12	15	2001	12/16/2001
7	15	12	31	2001	1/1/2002
8	16	2	15	2001	2/16/2001
9	17	2	28	2004	2/29/2004
10	18	2	28	2001	3/1/2001
11	19	2	29	2004	3/1/2004
12	20	2	29	2001	Invalid Input Date
13	21, 22	2	30	2001	Invalid Input Date



Cause and Effect Graphs (basic gates)



Cause and Effect Graph for the Commission Problem

