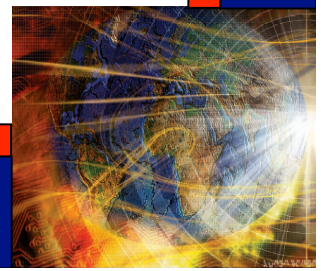
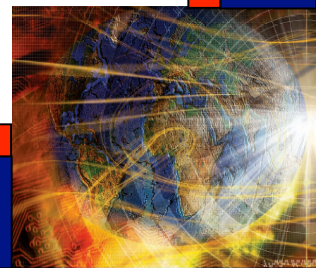
Chapter 7

Decision Table Based Testing



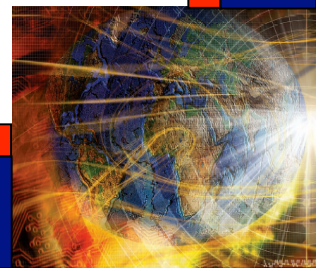
Decision Table Based Testing

- 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.

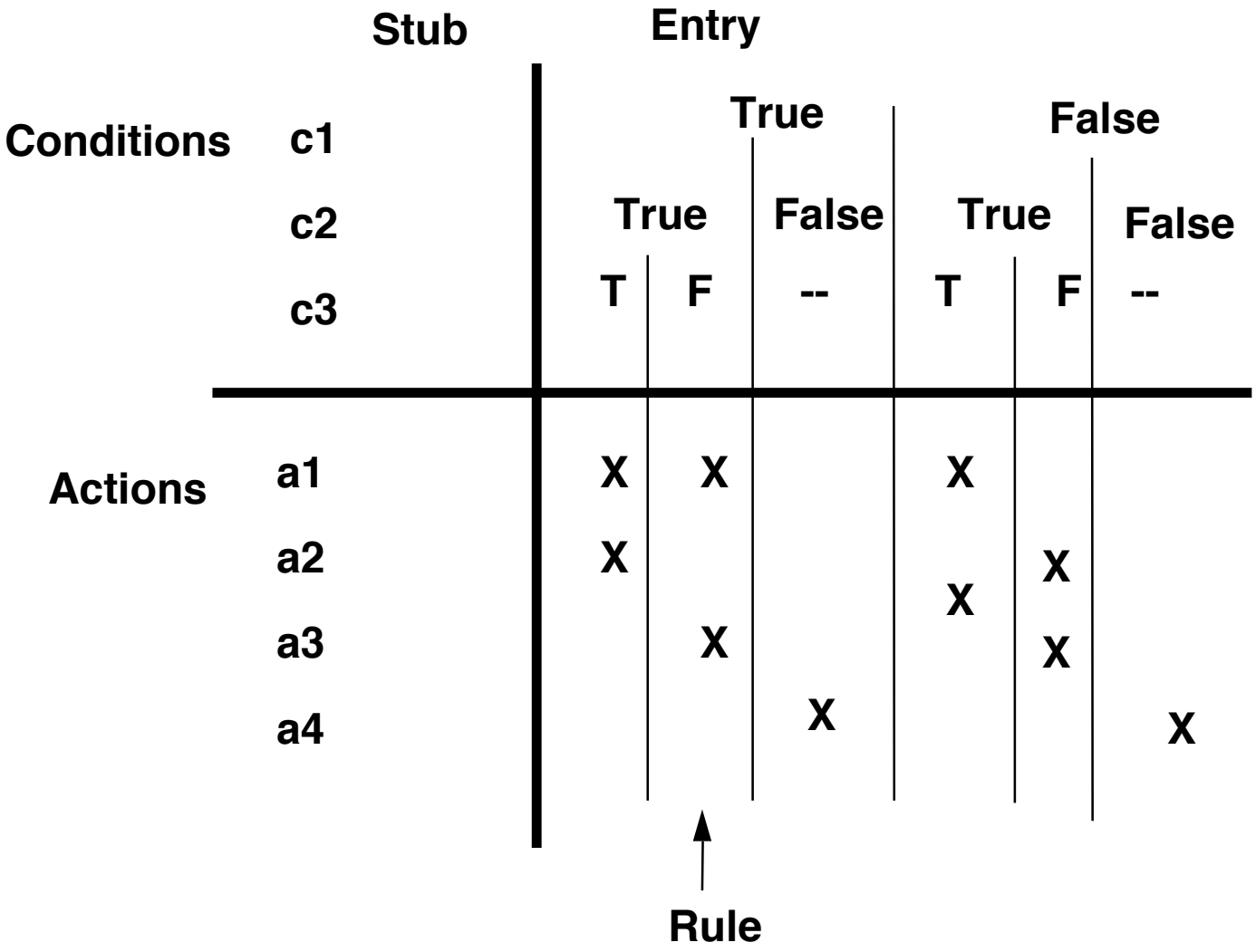


Decision Tables

- 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



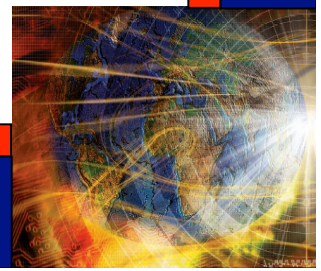
Decision Table Terminology



One Decision Table for the Triangle Problem

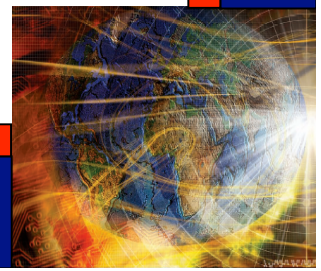
c1: a, b, c are a triangle?	N	Y	Y	Y	Y	Y	Y	Y	Y
c2: a = b?	--	Y	Y	Y	Y	N	N	N	N
c3: a = c?	--	Y	Y	N	N	Y	Y	N	N
c4: b = c?	--	Y	N	Y	N	Y	N	Y	N
a1: Not a triangle	X								
a2: Scalene									X
a3: Isosceles					X		X	X	
a4: Equilateral		X							
a5: Impossible			X	X		X			

Why are rules 3, 4, and 6 impossible?



Decision Table with Mutually Exclusive Conditions

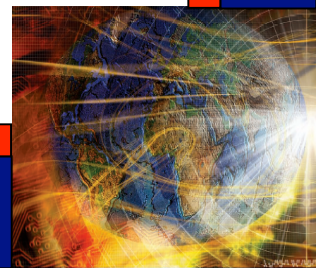
Conditions	Rule 1	Rule 2	Rule 3
c1: 30-day month	T	—	—
c2: 31-day month	—	T	—
c3: February	—	—	T



Rule Counting to Check for Completeness

conditions	R1	R2	R3
c1: month in M1	T	--	--
c2: month in M2	--	T	--
c3: month in M3	--	--	T
Rule count	4	4	4

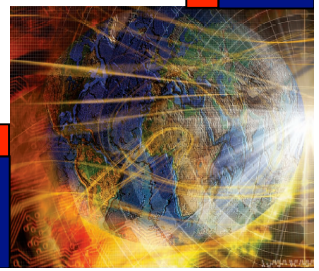
- A limited Entry decison table with n conditions has 2^n rules
- A Don't Care entry doubles the count of a rule
- What are the possibilities when rule count is not 2^n ?
- More precise to use F! (must be false) than -- (don't care)



A Redundant Decision Table

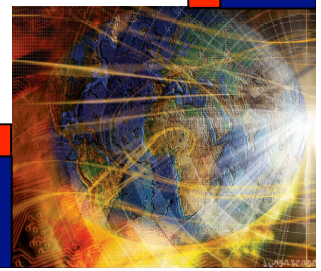
conditions	1-4	5	6	7	8	9
c1:	T	F	F	F	F	T
c2:	--	T	T	F	F	F
c3:	--	T	F	T	F	F
a1:	X	X	X	--	--	X
a2:	--	X	X	X	--	--
a3:	X	--	X	X	X	X

- Rule 9 is identical to Rule 4 (T, F, F)
- Since the action entries for rules 4 and 9 are identical, there is no ambiguity, just redundancy.



Decision Table Exercise
(revisit the false negative, false positive question)

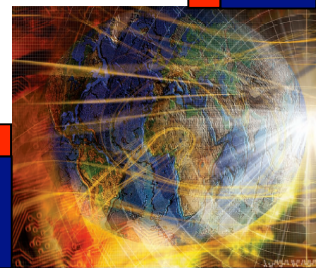
- **Suggested conditions**
  - Expected output is correct
  - Observed output is correct
  - Expected and observed outputs agree
- **Suggested actions**
  - True pass
  - True fail
  - False pass
  - False fail
  - Impossible



An Inconsistent Decision Table

conditions	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	--	--	--
a2:	--	X	X	X	--	X
a3:	X	--	X	X	X	--

- Rule 9 is identical to Rule 4 (T, F, F)
- Since the action entries for rules 4 and 9 are different there is ambiguity.
- This table is inconsistent, and the inconsistency implies non-determinism.



Nextdate Limited Entry Decision Table

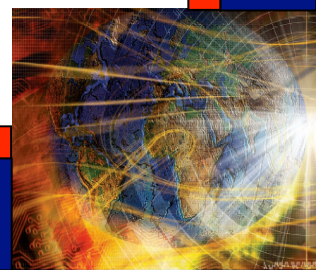
This decision table will have 256 rules, many of which will be logically impossible.

Conditions

c1: month in M1?
c2: month in M2?
c3: month in M3?
c4: day in D1?
c5: day in D2?
c6: day in D3?
c7: day in D4?
c8: leap year?

Actions

a1: impossible
a2: next date



Decision Table Based Test Cases

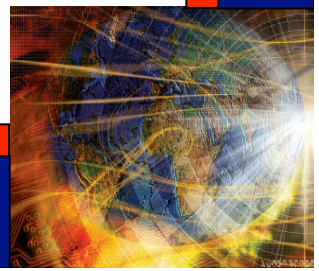
1. Decision table testing begins with equivalence classes for conditions as in equivalence class testing.
2. The sparseness due to the assumption of independence is addressed by careful examination of elements in the cross product.
3. For the equivalence classes defined earlier, the cross product contains 36 elements. The corresponding decision table has 36 rules.

<M1, D1, Y1>, <M1, D2, Y1>, <M1, D3, Y1>, <M1, D4, Y1>,
<M2, D1, Y1>, <M2, D2, Y1>, <M2, D3, Y1>, <M2, D4, Y1>,
<M3, D1, Y1>, <M3, D2, Y1>, <M3, D3, Y1>, <M3, D4, Y1>,

<M1, D1, Y2>, <M1, D2, Y2>, <M1, D3, Y2>, <M1, D4, Y2>,
<M2, D1, Y2>, <M2, D2, Y2>, <M2, D3, Y2>, <M2, D4, Y2>,
<M3, D1, Y2>, <M3, D2, Y2>, <M3, D3, Y2>, <M3, D4, Y2>,

<M1, D1, Y3>, <M1, D2, Y3>, <M1, D3, Y3>, <M1, D4, Y3>,
<M2, D1, Y3>, <M2, D2, Y3>, <M2, D3, Y3>, <M2, D4, Y3>,
<M3, D1, Y3>, <M3, D2, Y3>, <M3, D3, Y3>, <M3, D4, Y3>.

4. Notice that many of these are impossible, e.g., <M1, D4, * >



NextDate Extended Entry Decision Table

Conditions

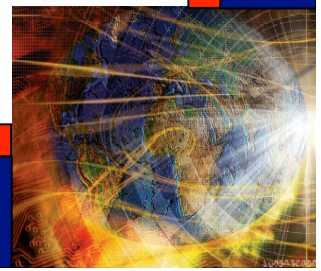
c1: month in M1? M2? M3?
c2: day in D1? D2? D3? D4?
c3: year in Y1? Y2? Y3?

Actions

a1: impossible
a2: increment day
a3: reset day
a4: increment month
a5: reset month
a6: increment year

This decision table will have 36 rules, and corresponds to the cross product. Many of the rules will be logically impossible.

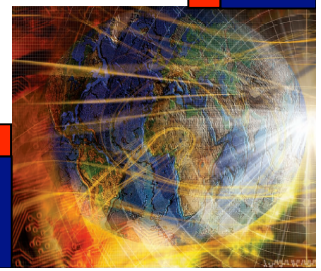
Many rules would collapse, except for considerations for December.



Revised NextDate Domain Equivalence Classes

- Month:
  - M1 = { month : month has 30 days}
  - M2 = { month : month has 31 days except December}
  - M3 = { month : month is December}
  - M4 = {month : month is February }
- Day
  - D1 = {day : 1 <= day <= 27}
  - D2 = {day : day = 28 }
  - D3 = {day : day = 29 }
  - D4 = {day : day = 30 }
  - D5 = {day : day = 31 }
- Year (are these disjoint?)
  - Y1 = {year : year is a leap year}
  - Y2 = {year : year is a common year}

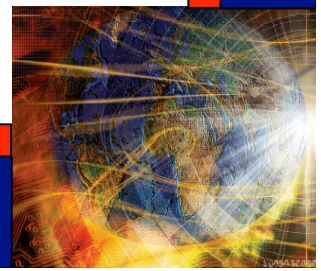
The corresponding decision table of these contains 40 elements.



NextDate Extended Entry Decision Table

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
c1: month in	M1					M2					M3					M4						
c2: day in	D 1	D 2	D 3	D 4	D 5	D 1	D 2	D 3	D 4	D 5	D 1,	D 2	D 3	D 4	D 5	D 1	D 2	D 2	D 3	D 3	D 4	D 5
c3: year in	-	-	-	-	-	-	-	-	-	-	-				-	-	Y 1	Y 2	Y 1	Y 2		-
a1: impossible					X																	
a2: increment day	X	X	X			X	X	X			X	X	X	X		X	X			X	X	X
a3: reset day				X					X						X			X				
a4: increment month				X						X								X	X			
a5: reset month										X					X				X			
a6: increment year															X							

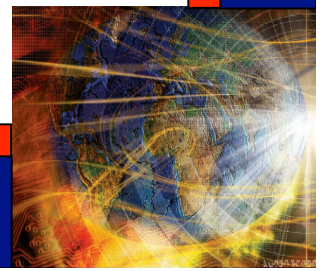
Notice there are 40 rules in this decision table, corresponding to the 40 elements in the cross product of the revised equivalence classes.



NextDate Extended Entry Decision Table

Algebraically Condensed to 13 rules (test cases)

rules	1-3	4	5	6-9	10	11-14	15	16	17	18	19	20	21,22
c1: month in	M1			M2		M3		M4					
c2: day in	D1,D2,D3	D4	D5	D1,D2,D3,D4	D5	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				
a3: reset day		X			X		X			X	X		
a4: increment month		X			X					X	X		
a5: reset month							X						
a6: increment year							X						

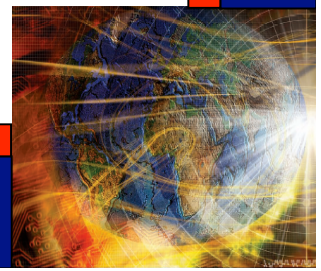


Procedure for Decision Table Based Testing

1. Determine conditions and actions. (Might need to iterate)

2. Develop a (the!) Decision Table, watching for
   - completeness
   - don't care entries
   - redundant and inconsistent rules

3. Each rule defines a test case.



Procedure for Decision Table Based Testing

- Determine conditions and actions. (Might need to iterate)
- Develop a (the!) Decision Table, watching for
  - Completeness
  - Don't care entries
  - Redundant and/or inconsistent entries
  - Impossible rules
- Each rule defines a test case.