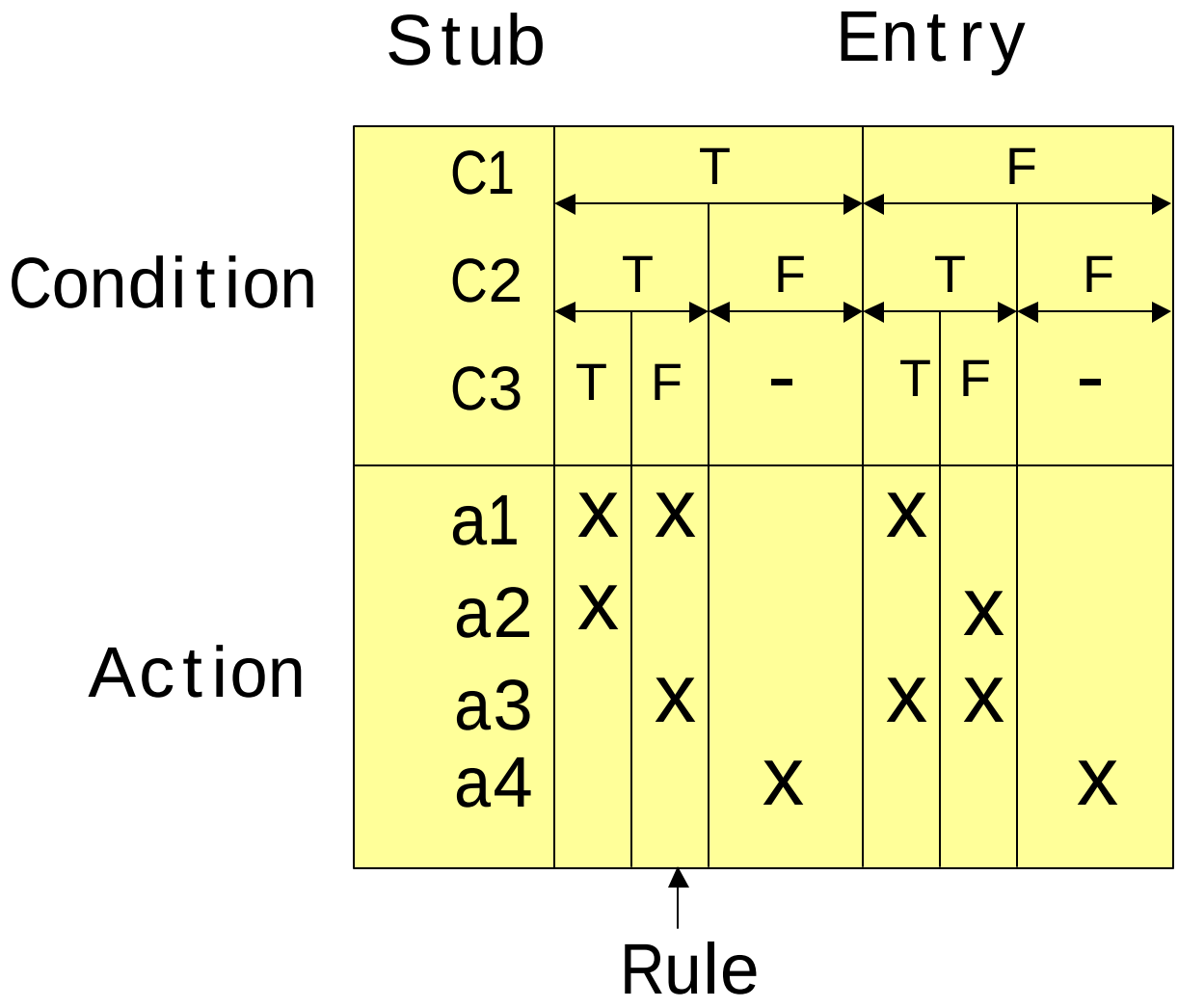
Decision Table Based Testing

Neelam Gupta

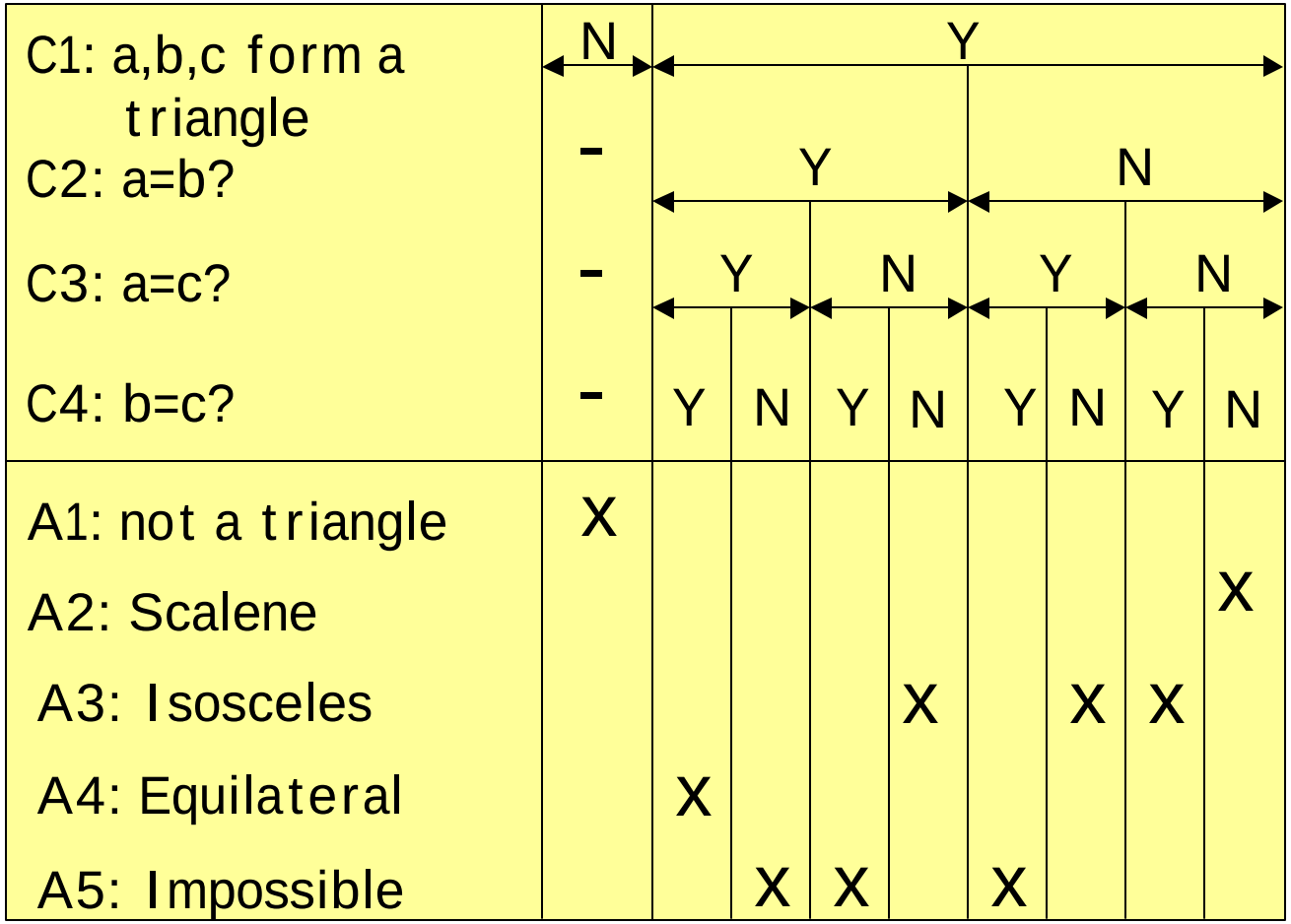
The University of Arizona
Tucson, Arizona, USA

Decision Tables



Test Case Generation

- Conditions: inputs
- Actions: outputs
- Rules: test cases



Triangle Problem: accept 3 integers as input

Output: Equilateral, Isosceles, Scalene, NotATriangle.

Expand c1: a,b,c are a triangle to a<b+c? b<a+c? c<a+b?

C1: a<b+c?	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
C2: b<a+c?		N	Y	Y	Y	Y	Y	Y	Y	Y	Y
C3: c<a+b?			N	Y	Y	Y	Y	Y	Y	Y	Y
C4: a=b?				Y	Y	Y	Y	N	N	N	N
C5: a=c?				Y	Y	N	N	Y	Y	N	N
C6: b=c?				Y	N	Y	N	Y	N	Y	N
A1: not a triangle	X	X	X								
A2: Scalene											X
A3: Isosceles							X		X	X	
A4: Equilateral				X							
A5: Impossible					X	X		X			

When conditions refer to equivalence classes, decision tables have characteristic appearance.

	R1	R2	R3
C1: month in M1	T	-	-
C2: month in M2	-	T	-
C3: month in M3	-	-	T
a1 a2 a3			

Rule count 2 for each don't care

4+4+4=12

Since we should have only 8 rules, where is the problem?

Condition	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4
C1: M1	T	T	T	T	T	T	F	F	T	T	F	F
C2: M2	T	T	F	F	T	T	T	T	T	F	T	F
C3: M3	T	F	T	F	T	F	T	F	T	T	T	T
Rule count	1	1	1	1	1	1	1	1	1	1	1	1

Delete repetitions, missing rule is in which all conditions are false.

Condition	1.1	1.2	1.3	1.4	2.3	2.4	3.4	
C1: M1	T	T	T	T	F	F	F	F
C2: M2	T	T	F	F	T	T	F	F
C3: M3	T	F	T	F	T	F	T	F
a1: impossible	x	x	x		x			x
a2:								
a3:								

The ability to recognize and develop complete decision tables makes them strong with respect to redundancy and inconsistency.

Condition	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	-

Rules 4 and 9 are inconsistent because action Sets are different. The whole table is non-deterministic.

Conclusion: Be careful with don't care entries used in a decision table.

Let us consider the decision table for next date problem with the following equivalence classes.

M1 = {month: month has 30 days}

M2 = {month: month has 31 days}

M3 = {month: month is February}

D1 = {day: 1<=day<=28}

D2 = {day: day=29}

D3 = {day: day=30}

D4 = {day: day=31}

Y1 = {year: year = 1900}

Y2 = {year: 1812<=year<=2012 AND (year!=1900) AND(year=0 mod 4)}

Y3 = {year: (1812<=year<=2012 AND year!=0 mod 4}

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C1: month in	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3	M3	M3	M3	M3
C2: day in	D1	D2	D3	D4	D1	D2	D3	D4	D1	D1	D1	D2	D2	D2	D3	D3
C3: year in	-	-	-	-	-	-	-	-	Y1	Y2	Y3	Y1	Y2	Y3	-	-
Rule Count	3	3	3	3	3	3	3	3	1	1	1	1	1	1	3	3
Actions																
a1: impossible				X								X		X	X	X
a2: increment day	X	X			X	X	X			X						
a3: reset day			X					X			?		X			
a4: increment month			X					?	X		?		X			
a5: reset month								?								
a6: increment year								?								

To resolve the ? entries, we need to refine the equivalence classes. Revised equivalence classes for the Next Date function are:

M1 = {month: month has 30 days}

M2 = {month: month has 31 days except December}

M3 = {month: month is December}

M4 = {month: month is February}

D1 = {day: 1<=day<=27}

D2 = {day: day=28}

D3 = {day: day=29}

D4 = {day: day=30}

D5 = {day: day=31}

Y1 = {year: year = 1900}

Y2 = {year: 1812<=year<=2012 AND (year!=1900)
AND(year=0 mod 4)}

Y3 = {year: (1812<=year<=2012 AND year!=0 mod 4}