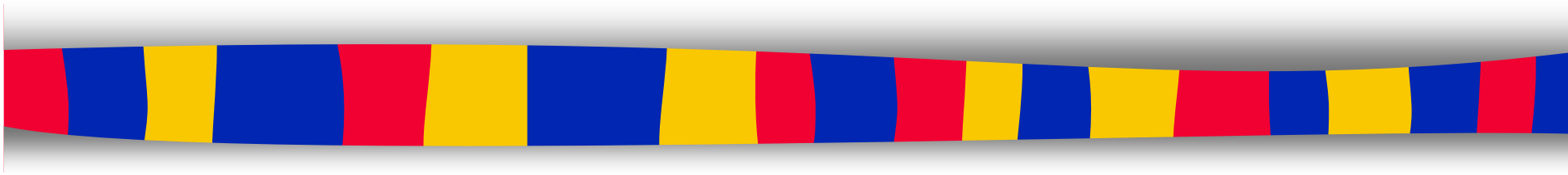
EECS 4313

Software Engineering Testing



Topic 06:

Decision Table-based Testing

Zhen Ming (Jack) Jiang

Relevant Readings

- [Jorgensen] chapter 7

Decision Tables - Wikipedia

- A precise yet compact way to model complicated logic
- Associate conditions with actions to perform
- Can associate many independent conditions with several actions in an elegant way

Decision Table Terminology

Stub	Rule 1	Rule 2	Rules 3,4	Rule 5	Rule 6	Rules 7,8
c1	T	T	T	F	F	F
c2	T	T	F	T	T	F
c3	T	F	-	T	F	-
a1	X	X		X		
a2	X				X	
a3		X		X		
a4			X			X

Condition stubs	condition entries
Action stubs	action entries

Decision Table Terminology

- Condition entries restricted to binary values
  - We have **limited entry table**
- Condition entries have more than two values
  - We have **extended entry table**

Printer Troubleshooting DT

Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognized	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

Let's try this for the Triangle problem

Triangle 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			

- The choice of conditions can greatly expand the size of a decision table.
- Need to have a more detailed view of the three inequalities of the triangle property (c1).
  - If any of the three fails, <a,b,c> won't constitute sides of a triangle

Refined Triangle Decision Table

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			

How to use decision table in software testing?

- Condition entries in a decision table are interpreted by a computer program as
  - input
  - equivalence classes of inputs
- Action entries in a decision table are interpreted as
  - output
  - major functional processing portions
- The rules are then interpreted as test cases.

Triangle Test Cases

Case ID	a	b	c	Expected Output
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

Don’t care entries and rule counts

- Limited entry tables with N conditions have 2^N rules
- Don’t care entries reduce the number of explicit rules by implying the existence of non-explicitly stated rules
  - Each don’t care entry in a rule doubles the count for the rule
  - For each rule determine the corresponding rule count
  - Total the rule counts

Refined Triangle Decision Table

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

When we add them up, it's 64 (2^6) rules

Count the rules in a decision table

- Less rules than combination rule count
  - Indicates missing rules
- More rules than combination rule count
  - Could indicate redundant rules
  - Could indicate inconsistent table

A example of 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

Which rule(s) is redundant?

A example of 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	-

Which rule(s) is inconsistent?

NextDate Decision Table

- The NextDate problem illustrates the problem of dependencies in the input domain
- Decision tables can highlight such dependencies
- Impossible dates can be clearly marked as a separate action
- Let’s try it…

NextDate 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 or 2100}
Y2= {year | year is a leap year}
Y3= {year | year is a common year}

NextDate Decision Table – mutually exclusive conditions

C1: month in M1?	T	-	-
C2: month in M2?	-	T	-
C3: month in M3?	-	-	T
A1: impossible			
A2: Next Date			

Because a month is an equivalence class, we cannot have T for more than one entry. The do not care entries are really “F”.

NextDate DT (1st try - partial)

C1: month in M1?	T	T	T	T	T	T	T	T	T	T	T	T
C2: month in M2?												
C3: month in M3?												
C4: day in D1?	T	T	T									
C5: day in D2?				T	T	T						
C6: day in D3?							T	T	T			
C7: day in D4?										T	T	T
C8: year in Y1?	T			T			T			T		
C9: year in Y2?		T			T			T			T	
C10: year in Y3?			T			T			T			T
A1: Impossible										X	X	X
A2: Next Date	X	X	X	X	X	X	X	X	X			

NextDate DT (2nd try - part 1)

C1: month in	M1	M1	M1	M1	M2	M2	M2	M2
C2: day in	D1	D2	D3	D4	D1	D2	D3	D4
C3: year in	-	-	-	-	-	-	-	-
A1: Impossible				X				
A2: Increment day	X	X			X	X	X	
A3: Reset day			X					X
A4: Increment month			X					?
A5: Reset month								?
A6: Increment year								?

NextDate DT (2nd try - part 2)

C1: month in	M3	M3	M3	M3	M3	M3	M3	M3
C2: day in	D1	D1	D1	D2	D2	D2	D3	D3
C3: year in	Y1	Y2	Y3	Y1	Y2	Y3	-	-
A1: Impossible				X		X	X	X
A2: Increment day		X						
A3: Reset day	X		X		X			
A4: Increment month	X		X		X			
A5: Reset month								
A6: Increment year								

New Equivalence Classes

M1= {month | month has 30 days}
M2= {month | month has 31 days, but not Dec.}
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 is a leap year}
Y2= {year | year is a common year}

NextDate DT (3rd try - part 1)

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 DT (3rd try - part 2)

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							

Decision Table Applicability

- The specification is given or can be converted to a decision table .
- The order in which the predicates are evaluated does not affect the interpretation of the rules or resulting action.
- The order of the rule evaluation has no effect on resulting action .
- Once a rule is satisfied and the action selected, no other rule need be examined.
- The order of executing actions in a satisfied rule is of no consequence.
- In reality, the restrictions do not eliminate many potential applications.
  - In most applications, the order in which the predicates are evaluated is immaterial.
  - Some specific ordering may be more efficient than some other but in general the ordering is not inherent in the program's logic.

Decision Tables - Issues

- Before deriving test cases, ensure that
  - The rules are complete
    - Every combination of predicate truth values is explicit in the decision table
  - The rules are consistent
    - Every combination of predicate truth values results in only one action or set of actions

Guidelines and Observations

- Decision Table testing is most appropriate for programs where
  - There is a lot of decision making
  - There are important logical relationships among input variables
  - There are calculations involving subsets of input variables
  - There are cause and effect relationships between input and output
  - There is complex computation logic (high cyclomatic complexity)

Guidelines and Observations (continued)

- Decision tables do not scale up very well
  - May need to
    - Use extended entry decision tables
    - Algebraically simplify tables
- Decision tables can be iteratively refined
  - The first attempt may be far from satisfactory
- Look for redundant rules
  - More rules than combination count of conditions
  - Actions are the same
  - Too many test cases
- Look for inconsistent rules
  - More rules than combination count of conditions
  - Actions are different for the same conditions
- Look for missing rules
  - Incomplete table