Decision Table-Based Testing

Chapter 7

Introductory question

- **What is a decision table?**
  - **A simple non-technical answer?**
  - **A detailed technical answer?**

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 – 2

- Condition entries restricted to binary values
  - We have a **limited entry table**

- Condition entries have more than two values
  - We have an **extended entry table**

condition stubs	**condition entries**
action stubs	**action entries**

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				

A complete limited entry table

Decision Table Interpretation

- **How are condition entries in a decision table interpreted with respect to a program?**

Decision Table Interpretation – 2

- Conditions are interpreted as
  - **Input**
  - **Equivalence classes of inputs**

Decision Table Interpretation – 3

- **How are action entries in a decision table interpreted with respect to a program?**

Decision Table Interpretation – 4

- Actions are interpreted as
  - Output
  - Major functional processing portions

Decision Table Interpretation – 5

- **What is the relationship between decision tables and test cases?**

Decision Table Interpretation – 5

- With a complete decision table
  - Have a complete set of test cases

Triangle Decision Table

	1	2	3	4	5	6	7	8	9
C1: <a, b,c > forms a triangle?	F	T	T	T	T	T	T	T	T
C3: a = b?	–	T	T	T	T	F	F	F	F
C4: a = c?	–	T	T	F	F	T	T	F	F
C5: 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**			

Action added by a tester showing impossible rules

Triangle Decision Table – refined

	1	2	3	4	5	6	7	8	9	10	11
C1-1: a < b+c?	F	T	T	T	T	T	T	T	T	T	T
C1-2: b < a+c?	–	F	T	T	T	T	T	T	T	T	T
C1-3: c < a+b?	–	–	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	X	X								
A2: Scalene											X
A3: Isosceles							X		X	X	
A4: Equilateral				X							
A5: Impossible					X	X		X			

Similar to equivalence classes we can refine the conditions

Triangle Test Cases

Case ID	a	b	c	Expected Output
1	4	1	2	Not a Triangle
2	1	4	2	Not a Triangle
3	1	2	4	Not a Triangle
4	5	5	5	Equilateral
5	???	???	???	Impossible
6	???	???	???	Impossible
7	2	2	3	Isosceles
8	???	???	???	Impossible
9	2	3	2	Isosceles
10	3	2	2	Isosceles
11	3	4	5	Scalene

Admission to University

- Students in
  - The range [80%, 100%] gpa are admitted and receive a scholarship.
  - The range [70%, 80%) gpa are admitted but have no scholarship.
  - The range [60%, 70%) gpa are admitted if they have no failures.
  - Otherwise they are not admitted.

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.
  - **How many rules does a table contain including all the implied rules due to don't care entries?**

Don't Care Entries and Rule Counts – 2

- 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

Don't Care Entries and Rule Counts – 3

1	C1-1: a < b+c?	F	T	T	T	T	T	T	T	T	T	T
2	C1-2: b < a+c?	–	F	T	T	T	T	T	T	T	T	T
3	C1-3: c < a+b?	–	–	F	T	T	T	T	T	T	T	T
4	C2: a = b?	–	–	–	T	T	T	T	F	F	F	F
5	C3: a = c?	–	–	–	T	T	F	F	T	T	F	F
6	C4: b = c?	–	–	–	T	F	T	F	T	F	T	F
	Rule count	32	16	8	1	1	1	1	1	1	1	1

+/ = 64

$= 2^6$

Don't Care Entries and Rule Counts – 4

- **How many rules do extended entry tables have?**
- **What is the rule count with don't care entries?**
  - See DDT-27..28 (NextDate 2'nd try)
  - See DDT-30-31 (NextDate 3'rd try)

Don't Care Entries and Rule Counts – 5

- **Is it useful to count the rules in a decision table?**
- **Why?**

Don't Care Entries and Rule Counts – 6

- Less rules than combination rule count
  - Indicates missing rules
- More rules than combination rule count
  - Could indicate redundant rules
    - See Table 7.9, page 107
  - Could indicate inconsistent table
    - See Table 7.10, page 108

NextDate Decision Table

- The NextDate problem illustrates the correspondence between equivalence classes and decision table structure
- 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

NextDate Equivalence Classes – for 1st try

M1 = {month : 1 .. 12 | days(month) = 30 }

M2 = {month : 1 .. 12 | days(month) = 31 }

M3 = {month : {2} }

D1 = {day : 1 .. 28}

D2 = {day : {29} }

D3 = {day : {30} }

D4 = {day : {31} }

Y1 = {year : 1812 .. 2012 | leap_year (year) }

Y2 = {year : 1812 .. 2012 | common_year (year) }

As in discussion for equivalence classes

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 in 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)

How many rules
- **for a complete table?**
- **with don't care entries?**

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

NextDate Equivalence Classes – for 2nd try

M1 = {month : 1 .. 12 | days(month) = 30 }

M2 = {month : 1 .. 12 | days(month) = 31 }

M3 = {month : {2} }

D1 = {day : 1 .. 28}

D2 = {day : {29} }

Handle leap year better

D3 = {day : {30} }

D4 = {day : {31} }

Y1 = {year : {2000} }

Y2 = {year : 1812 .. 2012 | leap_year (year) $\wedge$ year $\neq$ 2000 }

Y3 = {year : 1812 .. 2012 | common_year (year) }

NextDate DT (2nd try - part 1)

This table has 16 rules. How many rules for a complete table?

	1	2	3	4	5	6	7	8
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								???

Extended entry table – more refined actions

NextDate DT (2nd try - part 2)

	9	10	11	12	13	14	15	16
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 – for 3rd try

M1 = {month : 1 .. 12 | days(month) = 30 }

M2 = {month : 1 .. 12 | days(month) = 31 $\wedge$ month $\neq$ 12 }

M3 = {month : {12} }

M4 = {month : {2} }

D1 = {day : 1 .. 27}

D2 = {day : {28} }

D3 = {day : {29} }

D4 = {day : {30} }

D5 = {day : {31} }

Y1 = {year : 1812 .. 2012 | leap_year (year) }

Y2 = {year : 1812 .. 2012 | common_year (year) }

Handle end of month and year better

NextDate DT (3rd try - part 1)

A 22 rule table

	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 DT (3rd try - part 2)

	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							

Decision Table - Equivalence Class Comparison

- It has been shown that equivalence classes and decision tables can be closely related.
  - **What benefit do we get from using equivalence classes in place of decision tables?**
  - **What benefit do we get from using decision tables in place of equivalence classes?**

DT 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 rule evaluation has no effect on resulting action .
- Once a rule is satisfied and the action selected, no other rule need be examined.

DT Applicability – 2

- The order of executing actions in a satisfied rule is of no consequence.
- The restrictions do not eliminate many 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 Table Properties

- You have constructed a decision table
  - **Before deriving test cases, what properties should the decision table have?**

Decision Tables Properties – 2

- Before deriving test cases, ensure that
  - The rules are complete
    - Every combination of predicate truth values is 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 one or more of the conditions hold.
  - **What are those conditions?**

Guidelines and Observations – 2

- 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 is complex computation logic (high cyclomatic complexity)
  - There are cause and effect relationships between input and output – ???

Guidelines and Observations – 3

- **What are some problems with using decision tables?**

Guidelines and Observations – 4

- Decision tables do not scale up very well
  - May need to
    - Use extended entry decision tables
    - Algebraically simplify tables

- Decision tables need to be iteratively refined
  - The first attempt may be far from satisfactory
    - Similar to using equivalence classes

Guidelines and Observations – 5

- Look for redundant rules
  - More rules than combination count of conditions
  - Actions are the same
  - Too many test cases
  - See Table 7.9, page 107
- Look for inconsistent rules
  - More rules than combination count of conditions
  - Actions are different for the same conditions
  - See Table 7.10, page 108
- Look for Missing rules
  - Incomplete table