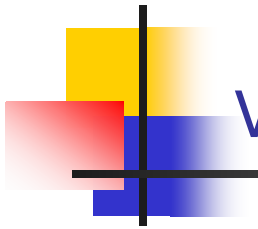
Variable Negation Strategy

Decision Table-Based Testing



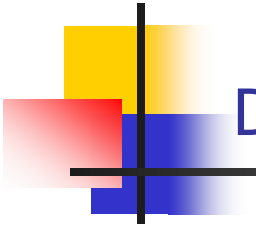
Variable Negation Strategy

- An approach that can help with the scaling problems of decision table-based testing
- Applicable when the system under test can be represented as a truth table (binary input and output)
- Designed to select a small subset of the 2^N test cases

Example truth table – Boiler controller

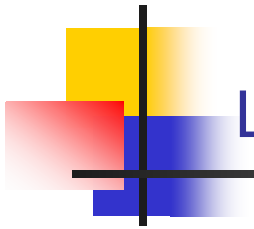
Z = F (A, B, C, D)

Variant Number	Normal Pressure	Call For Heat	Damper Shut	Manual Mode	Ignition Enable
	A	B	C	D	Z
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1



Deriving the Logic Function

- Boolean algebra expressions
  - **A B** ≡ A and B
  - **A + B** ≡ A or B
  - **~A** ≡ not A
    - ~A B C means ~A and B and C
    - ~(A B C) means ~A and ~B and ~ C
- A logic function maps N Boolean input variables to a Boolean output variable
- A truth table is an enumeration of all possible input and output values



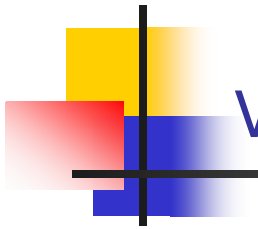
Logic function

- The logic function for the example is

  Z = A B ~C + A D

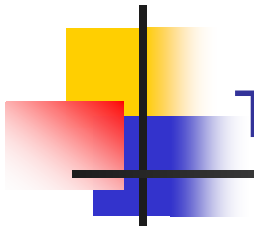
- Several techniques to derive it
  - Karnaugh maps
  - Cause-effect graphs

- A compact logic function will produce more powerful test cases



Variable Negation Strategy

- Designed to reveal faults that hide in a don't care
- The test suite contains:
  - **Unique true points**: A variant per term t, so that t is True and all other terms are False
    - In the expression **A B ~C + A D** , **A B ~C** and **A D** are terms
  - **Near False Points**: A variant for each literal in a term. The variant is obtained by negating the literal and is selected only if it makes Z = 0
- Each term variant creates a test candidate set



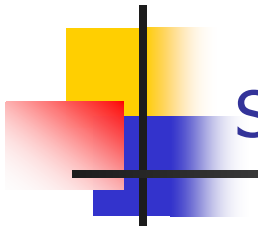
True points

- Unique true point candidate sets in boiler example
  - Variants in the set {12} make **A B ~C** true but not **A D**
    - Variant 13 makes both **A B ~C** and **A D** true and as a consequence is not included in the set
  - Variants in the the set {9,11,15} make **A D** true but not **A B ~C**
    - Variant 13 makes both **A B ~C** and **A D** true and as a consequence is not included in the set

Near false points

Candidate set number	Term negation	Function variants containing this negation	Function variants containing this negation where Z = 0
1 Org. term	**A B ~C**	**–**	**12**
2	A B C	14, 15	14
3	A ~B ~C	8, 9	8
4	~A B ~C	4, 5	4, 5
5 Org. term	**A D**	**–**	**9, 11, 15**
6	A ~D	8, 10, 12, 14	8, 10, 14
7	~A D	1, 3, 5, 7	1, 3, 5, 7

Near false points are in black, candidate set numbers 2, 3, 4, 6 and 7. In green are true points.



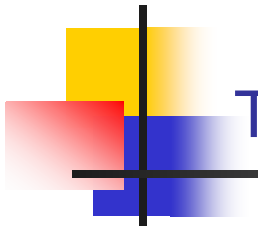
Selecting the test cases

- At least one variant from each candidate set
- Can be done by inspection
- Random selection is also used
- Near False Points exercise combinations of don't care values
- 6% of all possible tests are created
- 98% of simulated bugs can be found

Selecting test cases – 2

Test Candidate Set

Variant	1	2	3	4	5	6	7	Test case?
0								
1							X	
2								
3							X	
4				X				
5				X			X	X M
6								
7							X	
8			X			X		X M
9					X			M
10						X		
11					X			X .
12	X							X M
13								
14		X				X		X M
15					X			X .



Test suite

- Candidate sets

1	12
2	14
3	8
4	4, 5
5	9, 11, 15
6	8, 10, 14
7	1, 3, 5, 7

- Minimum Test suite variants

5	candidate sets 4 & 7
8	candidate sets 3 & 6
9	candidate set 5
12	candidate set 1
14	candidate set 2