

Black Box Testing

- Equivalence Classes
- Boundary Value Analysis
- Cause-Effect Graphing
- Error Guessing
 - Generating test cases bases on intuition

Equivalence Classes

- For each piece of the specification, generate one or more equivalence Class
- Label the classes as “Valid” or “Invalid”
- Generate one test case for each Invalid Equivalence class
- Generate a test case that Covers as many Valid Equivalence Classes as possible

Boundary Value Analysis

- Generate Equivalence Classes for BOTH input and output EC
- Generate test cases as specified in EC EXCEPT choose Boundaries for the test cases.
- Example: $1 \leq X \leq 100$
 - Use 1, 2, 99, 100 for valid EC
 - Use 0 and 101 for invalid EC.

Cause Effect Graphing

- Attempt to explore combinations of input conditions
- Technique which aids in selecting, in a systematic way, a high yield set of test cases
- **FORMAL** – language into which a natural language specification is translated.

Cause-Effect Graphing

1. Divide specification into small workable pieces
2. List
 - a. Causes – input Equivalence Classes
 - b. Effects – output Equivalence Classes
(Assign a unique number to each)
3. The Semantic content of the specification is analyzed and transformed into a Boolean graph linking the causes to the effects.

Cause-Effect Graphing

4. The graph is annotated with constraints describing **combinations** of causes and/or effects that are impossible due to **syntactic** constraints
5. By methodically tracing state conditions in the graph, the graph is converted to a limited entry decision table

Cause-Effect Graphing

- The columns in the decision table are converted into test cases.

Notation Identity

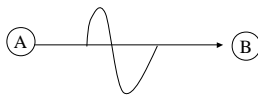
0 – absent state

1 – present state



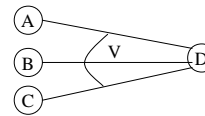
IF A = 1
THEN B = 1

Notation Not



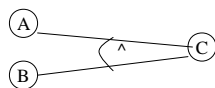
IF A = 1
THEN B = 0
ELSE B = 1

Notation OR



IF A = 1 OR B = 1 OR C = 1
THEN D = 1

Notation AND



IF A = 1 AND B = 1
then c = 1

Constraints