Summary of Fault-Based Adequacy Criteria

- Create tests to cover faults that could possibly occur in the software.
- Introduce mutations into the code.
- See if the tests detect the mutations.

Error Based Testing and Domain Analysis

- Divide the domain into subdivisions.
- Test inside and on the boundaries of the divisions.
- Two ways to subdivide
  - Specification-Based
  - Program-Based

Specification-Based Division

- Division by analyzing the problem's input.
- Example: Program converts a number from 0 to 100 to a letter grade (without + or -)
  - 83 and 84 both return B. Don't divide.
  - 89 returns B, 90 returns A. Divide here.

Program-Based Division

- Divisions are determined by looking directly at the code and identifying where the code branches.

```
char convertScoreToGrade(int
score) {
  if (score > 89) return 'A';
  else if (score > 79) return 'B';
  else if (score > 69) return 'C';
  else if (score > 59) return 'D';
  else return 'F';
}
```

Question

What are the advantages and disadvantages of each method?

Boundary Analysis

- N x 1 domain testing. Choose N test cases on the borders and 1 off. (White and Cohen, 1980)
- N x N domain testing. Choose N test cases on the borders and N linearly independent test cases off the border. (Clark et. al, 1982)
- N x 1 detects parallel boundary shifts.
- N x N adds rotation.

Boundary Analysis Example

- There are two input variables, so N is 2
- There is a boundary at numItems = 10 when costPerItem <= 100
- With N x 1, 2 tests would set numItems = 10 and costPerItem <= 100 (on boundary), 1 test would set numItems <> 10 and costPerItem <= 100 (off boundary)
- N x N would be the same as N x 1, but there would be 2 tests of the second type instead of 1.

```
int calculateTotalCost(int numItems,
    int costPerItem) {

  int totalCost = numItems *
costPerItem;

  if ((costPerItem > 100
       && numItems > 5)
    || (numItems > 10)) {
    totalCost *= 0.9;
  }

  return totalCost;
}
```

Functional Analysis

- Boundary analysis looks for errors in boundary location. Functional analysis looks for errors in computation inside each division.
- Example: total = 5*x + 10*y
  - f(x, y) = 5x + 10y
  - f is a linear function, and two points define a line, so two test cases are required.
  - For the two examples, x and y would be set to any values that are in the division, for example if this function calculated the output for the division of a program where x < 10, x and y would be set to any arbitrary values as long as x is less than 10.

Question

What are some limitations of domain analysis?

Comparison of Test Data Adequacy Criteria

- Three types of comparison appear in the literature:
  - Fault-Detecting Ability
  - Software Reliability
  - Test Cost

Fault-Detecting Ability

- Statistical Experiment
- Simulation
- Formal Analysis

Statistical Experiment

- Choose a set of programs with known faults (either through previous experience or mutations)
- A test set is generated using some method
- The proportion of faults detected compared with the known number of faults is calculated
- Statistical analysis is done on these numbers

Statistical Experiment Example

- Test set C1 detects 17 of 41 faults on its program, while test set C2 detects 80 of 200 faults.
- The proportion of faults detected by C1, p1, is .415. For C2, p2 = 0.400.
- C1 appears slightly better, but statistically p1 has a greater margin of error.

Simulation Method

- Generate random test cases and run them on a set of programs
- Research by Duran and Ntafos showed that 100 simulated random test cases performed better than 50 simulated partition test cases.

Formal Analysis

- Compares test sets by formally proving relations between them
- Five relations:
  - C1 narrows C2
  - C1 covers C2
  - C1 partitions C2
  - C1 properly covers C2
  - C1 properly partitions C2

Software Reliability

- The adequacy of a test set can be directly measured by the reliability of the software that passed the tests.
- Methods exist to measure the reliability of software.

Test Cost

- The third way to compare test adequacy is by the test cost.
- Because testing is expensive, it is important to consider the cost of a test method.
- If a method is only slightly better than another one, but much more expensive, it may not be the best choice.
- What types of things would be considered when determining test cost?

Summary of Comparison of Test Adequacy

- It is easier to achieve high confidence using partition testing
- Random testing is cheaper
- Is one of these methods always better? How would you decide which method to use in a given situation?

Axiomatic Assessment

- Seek the most fundamental properties of test adequacy
- An axiomatic approach has been proven useful in math and physics

Example of Axioms (Weyuker)

- A1 (Applicability): For every program, there exists a finite adequate test set.
- A2 (Nonexhaustive Applicability): There is a program p and a test set t such that p is adequately tested by t and t is not an exhaustive test set.
- A3 (Monotonicity): If t is adequate for p and t is a subset of t', then t' is adequate for p.
- A4 (Inadequate Empty Set): The empty set is not adequate for any program.

Conclusion

- Test criteria are a central problem of software testing.
- Numerous adequacy criteria have been proposed, analyzed and compared.
- Much research has also been done on the issue of evaluating and comparing criteria.
- The tendency is towards systematic approaches in testing using adequacy criteria.