COMP 3711

(OOA and OOD)

Software Testing 3
Equivalence Class

Testing Computer Software (2ed) Kaner/Falk/Nguyen 7

Equivalence Classes

- If two tests produce the same result, they’re equivalent
- A group of test forms an equivalence class if:
  - They all test the same thing
  - If one test finds a bug, the others probably will too
  - If one test misses a bug, the others probably won’t

Indications of the Same Equivalence Classes

- They involve the same input variable
- They result in similar operations in the program
- They affect the same output variable
- None of them force the program to do error handling or they all do

Finding Equivalence Classes

- Look at invalid inputs
  - For example: A program that accepts numbers from 1 to 99 as input, there are 4 equivalence classes:
    - Correct input
    - Numbers less than 1
    - Numbers greater than 99
    - Inputs that are not numbers
- Organize classifications into a table or an outline

Building a Table

- Tabular format is easier to read
- Easier to distinguish between equivalence classes for valid and invalid inputs
- Easier to evaluate coverage of invalid equivalence classes
- Unfortunately tables can become very large

Example: Table Listing Equivalence Classes

Input or Output Event	**Valid Equivalence Classes**	**Invalid Equivalence Classes**
Enter a number	Numbers between 1 and 99	0 > 99 An expression that yields an invalid number, such as 5-5, which yields 0 Negative numbers Letters and other non-numeric characters
Enter the first letter of a name	First character is a capital letter First character is a lower case letter	First character is not a letter
Draw a line	From 1 dot-width to 4 inches long	No line Line longer than 4 inches Not a line (curve)

Build An Outline

- Good outline processor makes it easy to add to, change, reorganize, reformat and print the outline
- Can break down classes and conditions more finely than with a table
- Don’t be concerned about overlapping equivalence classes
- Programmers tend not to spend much time testing invalid inputs

Example: Outline For Equivalence Classes

1. Enter a number

1.1 Valid Case

1.1.1 Between 1 and 99

1.2 Invalid Cases

1.2.1 0

1.2.2 > 99

1.2.3 A calculation whose result is invalid such as 5 - 5 yielding 0

1.2.4 Negative numbers

1.2.5 Letters and other non-numeric characters

1.2.5.1 Letters

1.2.5.2 Arithmetic operators such as +, *,-

1.2.5.3 The rest of the non-numeric characters

1.2.5.3.1 Characters with ASCII codes below the code for 0

1.2.5.3.2 Characters with ASCII codes above the code for 9

Finding Equivalence Classes

- **Look for membership in a group**
  - If a country name is required as input, all valid country names are one equivalence class
  - Everything else belongs to another class
  - May be able to subdivide them

Continue …

- **Analyze responses to lists and menus**
  - Program responds differently to each item
    - Yes/no
  - Each input is an equivalence class
  - Invalid everything not on list? None?

Continue …

- **Look for variables that must be equal**
  - Only one possible input is allowed
  - All other inputs have become invalid and now belong to their own equivalence class
    - e.g. everything else sold out

Continue …

- **Create time-determined equivalence classes**
  - Things done *long before* the task is done are one equivalence class
  - Everything you do *just before* the program starts is another equivalence class
  - Everything you do *while* the program is done is a third equivalence class
    - e.g. printer

Continue …

- **Look for variable groups that must calculate to a certain value or range**
  - For example - testing the sum of angles:
    - If you’re entering the three angles for a triangle, they must sum to 180^{o}
    - All sets of angles that sum to less than 180^{o} belong to another equivalence class
    - All sets of angles that sum to more than 180^{o} belong to a third equivalence class

Continue …

- **Look for equivalent output events**
  - What classes of inputs produce same outputs
  - What inputs will produce different types of error output?
  - Equivalence classes for inputs and outputs won’t necessarily correspond

Continue …

- Example: Specification says that after several computations a number between 1 and 45 will be printed.
  - What input would make it print something less than 1?
  - Greater than 45?
  - Create test cases to try them

Continue …

- **Look for equivalent operating environments**
  - Programs may be affected by the speed of the processor, the amount of memory, the available disk space
  - For example: The program will work if computer has between 64 and 256K of memory
    - Also clock speed, number of peripheral hardware …

Boundaries of Equivalence Classes

- Normally only use one or two test cases from an equivalence class
- Boundary values are the best ones
  - Generally most extreme values in class
- Programs that fail with non-boundary values usually fail at the boundaries too
  - (but not the other way around)
- Input boundary values do not necessarily generate output boundary values
  - Important to test both