FIT2107
Software Quality and Testing

Lecture 4–Black-Box Testing II

- Boundary Value Testing
- Combinatorial Testing
  - Pairwise Testing

Dr. Najam Nazar

Recap

- Random testing
- Equivalence testing, and
- Category Partitioning

Let’s go through some examples first to recap what we had studies last week.

Exercise : Monash letter grader

- Single Python (or Javascript) function.
- (Valid) input: one integer *mark* [0, 100].
- Expected output: appropriate letter grade.

Grade	Description	Mark
HD	High Distinction	80–100
D	Distinction	70–79
C	Credit	60–69
P	Pass	50–59
N	Fail	0–49

- What are the representative categories?
  - Do they have homogenous behaviour?
  - Are these 5 categorise sufficient?
  - How about invalid inputs?

- What are the representative values?
  - For Grade D – anything from [70-79]?

MONASH University

Exercise : Monash letter grader

- Category: Grade Type (HD, D, ….)
- Choices: number, string, other object.
- Should our function handle string of digits as numbers?
- If we were writing in a statically-typed language (e.g. Java), we wouldn't care about this for an internal-facing function…
- Do we need to differentiate between "other objects"?
- What about numerical grades <0, >100?
- Maybe add "invalid (too low)", "invalid (too high)"

Exercise : Monash letter grader

Test Frames

Identified test frames

1. Type = number, Grade = HD {80-100}
2. Type = number, Grade = D {70-79}
3. Type = number, Grade = C {60-69}
4. Type = number, Grade = P {50-59}
5. Type = number, Grade = N {0-49}
6. Invalid – too low {<0}
7. Invalid – too high {>100}
8. Type = string, Grade = HD {“80” - “100”}

……

Test frames

- Theoretically, we need to test each and every combination
- But we need to strive a balance between maximizing the test coverage and minimizing the cost (monetary and effort) of testing
- Most combinations are invalid (that is, they CAN'T exist, rather than they result in an execution error). (i.e. user provide erroneous inputs)
- Valid combinations, "Type = number" with all the grades…
- You can add a "N/A" partition to the "grade" choice if it helps you work through the possibilities.

Exercise : Monash letter grader

Turning test frames into test cases

Type = number, "Grade = P"

Pick an input where the type is a number and it should result in the output being "P".

Mark=53 is a *test input*

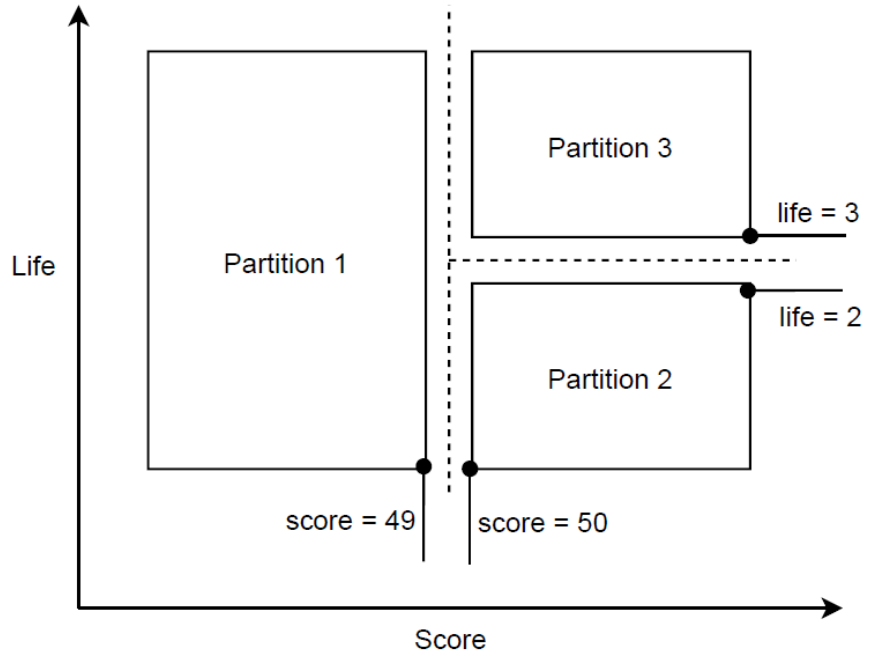
Mark=53, expected output is "P", is a test case.

Boundary Value Partitioning (Analysis)

- A high number of bugs happen in the boundary
- Some terminologies
  - **On-point:** the value that is on the boundary.
  - **Off-point:** the value closest to the boundary that flips the conditions
  - **In-point:** are all the values that make the condition true
  - **Out-point:** are all the values that make the condition false

Boundary Value Partitioning (Analysis)

- **Example:** Given the score of a player and the number of remaining lives of the player, the program does the following:
  - If the player's score is below 50, then it always adds 50 points on top of the current points.
  - If the player's score is greater than or equals to 50, then:
    - if the number of remaining lives is greater than or equal to 3, it triples the score of the player.
    - otherwise, it adds 30 points on top of the current points.
- Partitions?
  - Less points:
    - Score < 50
  - Many points but little lives:
    - Score >= 50 and remaining lives < 3
  - Many points and many lives:
    - Score >= 50 and remaining lives >= 3

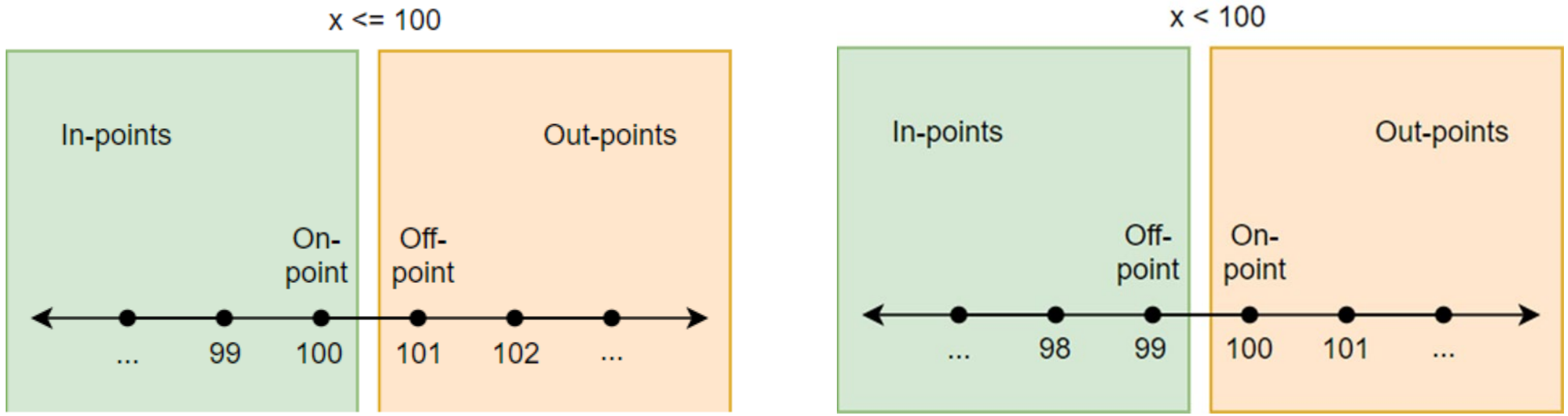


Boundary Value Partitioning (Analysis)

- Partitions?
  - For B1:
    - B1.1 = input= {score=49, remaining lives=5}, output= {99}
    - B1.2 = input= {score=50, remaining lives=5}, output= {150}
  - For B2:
    - B2.1 = input= {score 500, remaining lives=3}, output= {1500}
    - B2.2 = input= {score 500, remaining lives=2}, output= {530}

Boundary Value Partitioning (Analysis)

- **Example 2:** Suppose we have a program that adds shipping costs when the total price is below 100.
  - The condition used in the program is x < 100.
  - The on point is 100, as that is the value in the condition.
  - The on-point makes the condition false, so the off-point should be the closest number that makes the condition true. This will be 99, 99 < 100 is true.
  - The in-points are all the values smaller than or equal to 99. For example, 37, 42, 56.
  - The out-points are all values larger than or equal to 100. For example, 325, 1254, 101.
- Partitions?



Combinatory Testing

- A method that can reduce the cost
- Improve the test effectiveness
- Not every parameter contributes in failure.
- Software failures are caused by interactions between relatively few parameters.

Pairwise combinatoric testing

- In pairwise testing, we design test cases to execute all possible discrete combinations of each pair of input parameters

Problem: Display control of the system

Inputs:

  - **Display mode:** graphics, text only and limited bandwidth.
  - **Supported languages:** English, French, Mandarin and Arabic.
  - **Fonts:** minimal, standard and document loaded.
  - **Colours:** monochrome, colormap, 16-bit, True colour
  - **Screen-Size:** Laptop, Full Size, Handheld.

- If we were to test the system exhaustively, we will need 3*4*3*4*3 = 432 possible test.
- Using pairwise testing we can reduce the number of test cases.

Tests	Combinations
1	English,Monochrome,Full-graphics,Minimal,Hand-held
2	English,Color-map,Text-only,Standard,Full-size
3	English,16-bit,Limited-bandwidth, -, Full-size
4	English,True-color,Text-only,Document-loaded,Laptop
5	French,Monochrome,Limited-bandwidth,Standard,Laptop
6	French,Color-map,Full-graphics,Document-loaded,Full-size
7	French,16-bit,Text-only,Minimal, -
8	French, True-color, -, -, Hand-held
9	Arabic,Monochrome, -, Document-loaded, Full-size
10	Arabic, Color-map, Limited-bandwidth, Minimal, Hand-held
11	Arabic, 16-bit, Full-graphics, Standard, Laptop
12	Arabic, True-color, Text-only, -, Hand-held
13	Mandarin,-, -, Monochrome, Text-only
14	Mandarin, Color-map, -, Minimal, Laptop
15	Mandarin, 16-bit, Limited-bandwidth, Document-loaded, Hand-held
16	Mandarin, True-color, Full-graphics, Minimal, Full-size
17	Mandarin, True-color, Limited-bandwidth, Standard, Hand-held

Example 2

Problem: Platform configuration parameters

Inputs:

- **OS:** Windows XP, Apple OS X, Red Hat Linux.
- **Browser:** Internet Explorer, Firefox.
- **Protocol:** IPv4, IPv6.
- **CPU:** Intel, AMD.
- **DBMS:** MySQL, Sybase, Oracle.

- If we were to test the system exhaustively, we will need 3*2*2*2*3 = 72 possible test.
- Do we need 72 Test Cases?

Tests	OS	Browser	Protocol	CPU	DBMS
1	XP	IE	IPV4	Intel	MySQL
2	XP	Firefox	IPV6	AMD	Sybase
3	XP	IE	IPV6	Intel	Oracle
4	OS X	Firefox	IPV4	AMD	MySQL
5	OS X	IE	IPV4	Intel	Sybase
6	OS X	Firefox	IPV4	Intel	Oracle
7	RHL	IE	IPV6	AMD	MySQL
8	RHL	Firefox	IPV4	Intel	Sybase
9	RHL	Firefox	IPV4	AMD	Oracle
10	OS X	Firefox	IPV6	AMD	Oracle

- Only 10 test needed, if we want to test all interactions of one parameter with one other parameter (pairwise interaction).
- # of pairs: $\frac{5!}{2!(5-2)!} = 10$

Jenny Tool

- Command-line tool for generating combinatorial tests.
- Need to tell?.
  - Pairwise, 3-wise…n-wise
- How many "dimensions"
- How many values/categories in each dimension.

```
xubuntu@xubuntu-VirtualBox:~/teaching/4004/tools/jenny$ ./jenny -n2 2 2 2 2 2 2
2 2 2
 1a 2a 3b 4a 5b 6a 7b 8b 9b
 1b 2b 3a 4b 5a 6b 7a 8a 9a
 1a 2b 3b 4b 5a 6a 7a 8b 9b
 1b 2a 3a 4a 5b 6b 7b 8a 9a
 1a 2b 3a 4a 5a 6b 7b 8b 9a
 1b 2a 3a 4b 5b 6a 7a 8a 9b
 1b 2b 3b 4a 5b 6b 7a 8a 9b
 1b 2a 3b 4b 5a 6a 7b 8b 9a
 1a 2a 3a 4b 5b 6a 7a 8a 9b
```

How do we know if our testing is adequate?

- Hard with pure black-box testing.
- Should trace test cases back to requirements!
- Is one test per requirement enough???
  - Usually not.
- But how many tests for each requirement?
  - If they're natural language requirements, too vague for a computer to tell you
- So…what do we do?
  - Wait for the section on white-box testing?

Summary

- We applied a number of black-box testing methods
- Chose based on nature of requirement.
- Always look for extra tests after applying techniques.
- Trace back to requirements.
- Problem: how many tests are enough? No good answer…yet!

Questions

MONASH University