Metamorphic Testing

Tsong Yueh Chen

tychen@swin.edu.au
Swinburne University of Technology
Australia

Basic Concepts

- Input domain – set of all possible inputs
- Failure-causing inputs – inputs revealing failure

Example

Consider the following program

```
Input A, B   / $0 \le A, B \le 10$
C = A+B      / should be C=A*B
Output C
```

Example (continued)

(A=2.5, B=1.8) is a valid input

(A=2.5, B=1.8) is a failure-causing input

(A=2.0, B=2.0) is not a failure-causing input

- Reliable Test Set Problem

  various test case selection strategies

- Test Oracle Problem

Solving a system of linear equations

$$x + y - z = 1$$

$$2x - y + z = 3$$

$$x + 3y + z = 10$$

Outputs: $x=1$, $y=2$ and $z=3$

Verifying the outputs

- Substitution

  $x=1$, $y=2$ and $z=3$

  $x + \mathrm{y} - \mathrm{z} = 1$

- Evaluation

  $1 + 2 - 3 \quad ? \quad 1$

Test Oracles

- Mechanisms or procedures against which the computed output for any input could be verified

Test Oracles (continued)

- A backward substitution
- An inverse function
- ……..
- ……..

Example 1

- *sin* function
  - *sin*(0^{o})=0
  - *sin*(30^{o})=0.5
- Suppose the program returns:

  sin(29.8^{o})=0.51234 incorrect

  sin(29.8^{o})=0.49876 correct?

Example 2

- Shortest Path $SP(G, a, b)$
- Suppose the program returns:
  - $|SP(G, a, a)| = 5$ incorrect
  - $|SP(G, a, b)| = 10$ where a and b are neighbours
  - $|SP(G, a, b)| = 123{,}456{,}789$ correct or incorrect?

Test Oracle Problem

- Absence of test oracle
- Test oracle too expensive to apply

Test oracle problem is common

A simple yet effective method to alleviate the test oracle problem

Metamorphic Testing: Intuition

Though we are not able to verify the correctness of the output for every individual input

we may know the relation between some related inputs and their outputs

Example 1

- Suppose *sin*(29.8^o) returns 0.49876
- *sin* function has the following properties
  - – if y=x+360 then *sin*(y)= *sin*(x)
- Compute $29.8^o + 360^{o\,=} 389.8^o$
- Execute the program using 389.8^o
- Check whether *sin*(29.8^o) = *sin*(389.8^o)

Metamorphic Testing (A Simplified Form)

- Define source (initial) test cases using some test case selection strategies
- Identify some properties of the problem (referred to as the metamorphic relations)
- Construct follow-up test cases from the source test cases with reference to the identified metamorphic relations
- Verify the metamorphic relations against the computed results

Example 1

- Suppose *sin*(29.8^o) returns 0.49876

  29.8^o is the source test case

- *sin* function has the following property
  - ***“If y=x+360 then sin*(y)= *sin*(x)” is an MR**

- Compute 29.8^o + 360^o = 389.8^o

  389.8^o is the follow-up test case

- Execute the program with 389.8^o as input
- Check whether *sin*(29.8^o) = *sin*(389.8^o)

For MR : If $y=x+360$ then $\sin(y)= \sin(x)$

<29.8, 389.8> is one of its metamorphic test groups

Example 2

- Shortest Path $SP(G, a, b)$ has the following MRs:
  - $|SP(G, a, b)| = |SP(G, b, a)|$
  - $|SP(G, a, b)| = |SP(G, a, c)| + |SP(G, c, b)|$ where c is a node in $SP(G, a, b)$

Metamorphic Testing

- Simple
- Straightforward implementation
- Ease of automation given the availability of MRs
- Low costs

Metamorphic Testing

- Some reminders
  - MRs not restricted to identity relations and numeric relations
  - Multiple executions
  - Follow-up test cases may depend on the outputs of the source test cases
  - MT is applicable even if test oracle exists

Successful Applications of MT

- Bioinformatics – *BMC Bioinformatics*, 2009
- GCC and LLVM C Compilers (EMI) – *PLDI*, 2014
- NASA DAT systems – *ICSE*, 2015
- Obfuscators: Tigress, Cobfusc, Stunnix CXX-Obfus – *Computer*, 2016

Successful Applications of MT (continued)

- Graphics shader – (GraphicsFuzz, a spin off company from Imperial College, acquired by Google in August, 2018)
- Self-driving cars – “Metamorphic Testing of Driverless Cars” by Zhou et al. to appear in *CACM*

Application Domains

- Prediction systems
  - weather forecasting, earthquake prediction
- AI systems
  - Machine learning: Accenture
- Simulations
  - Epidemiological model (Oak Ridge National Lab)
  - Storm water management model (US Environmental Protection Agency)
- End-user programming

Identification of MRs

- Is it feasible to identify or generate MRs?

A Popular and Intuitive Approach

- Select an input
- Modify it, hopefully that the relevant change of output will be somehow predictable.

If yes, any generalisation?

If yes, then identify an MR

Example 1

To find the sum of a series of integers

Input is: {3, 7, 12, 6, 8, 6, 3, 5, 15, 4}

What are the possible MRs?

What are the possible follow-up test cases?

Example 2

To sort a series of integers which may be duplicated, in ascending order without duplications

Input is: {3, 7, 12, 6, 8, 6, 3, 5, 15, 4}

What are the possible MRs?

Example 6

To search how many times and where a string (S1) appears in another string (S2)

S1 is abac

S2 is abaccabacddaabcabdcdaabacccdbabaddc

*abac*c*abac*ddaabcabdcda*abac*ccdbabaddc

Example 6 (continued)

Note: (S1, S2) form an input

Possible follow-up test cases (S1’, S2’) are

- (Sl’, S2’) where S1’=S1, S2’=S2+S2
- (Sl’, S2’) where S1’=S1+S1, S2’=S2
- (Sl’, S2’) where S1’ are S2’ are from S1 and S2 after the same permutation scheme

Two Important Testing Results of MT

- GCC and LLVM C Compilers (EMI)
- Siemens suite
  - print_token, schedule, and schedule_2

Diversity

underlying concept in software testing

Conclusion

Thanks!

References:

- A Survey on Metamorphic Testing, S. Segura, G. Fraser, A. B. Sanchez and A. Ruiz-Cortes, *IEEE Transactions on Software Engineering*, Vol. 42(9),805-824, 2016.
- Metamorphic Testing: A Review of Challenges and Opportunities, T. Y. Chen, F.-C. Kuo, H. Liu, P. L. Poon, D. Towey, T. H. Tse and Z. Q. Zhou, *ACM Computing Surveys*, Vol. 51(1), 4:1-4:27, 2018.