Software Testing Techniques

Example related to Regression Testing.

Two possibilities, i) when oracle is available (example is in next slide) and ii) when the oracle is not available or is available but too expensive to apply.

Oracle problem

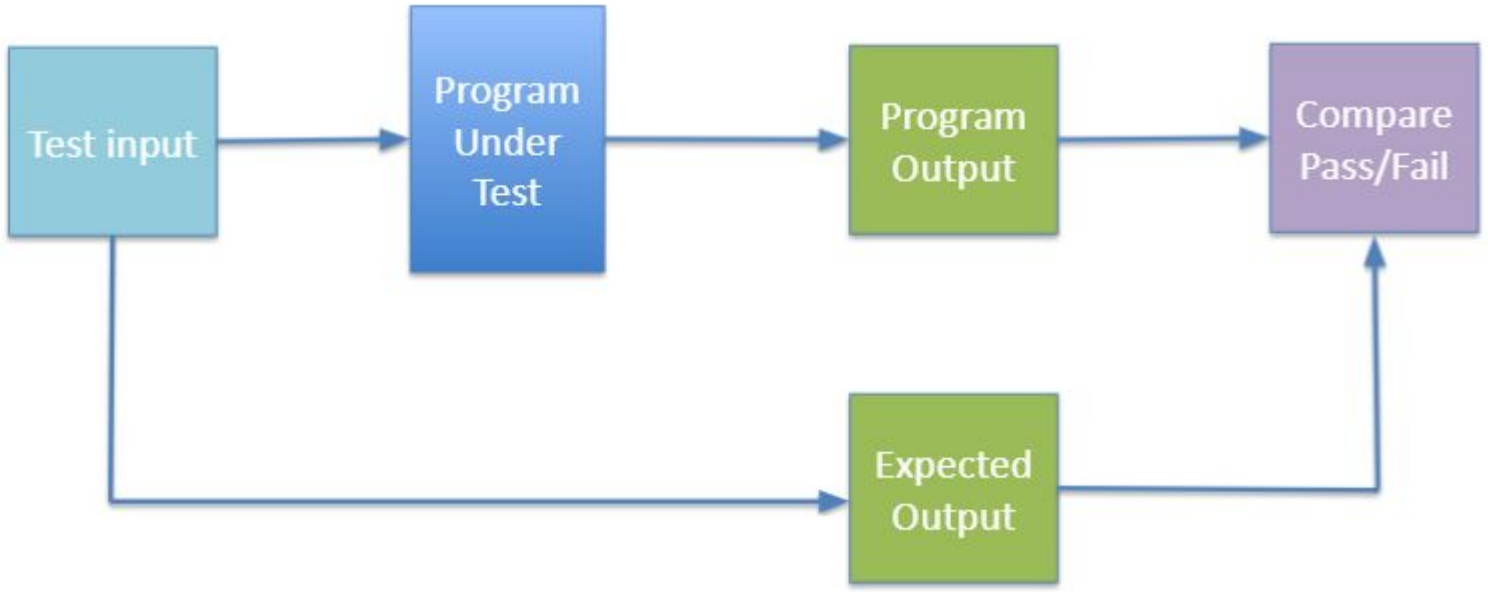
Couple of popular approaches that can be used to alleviate oracle problem.

i) Differential Testing

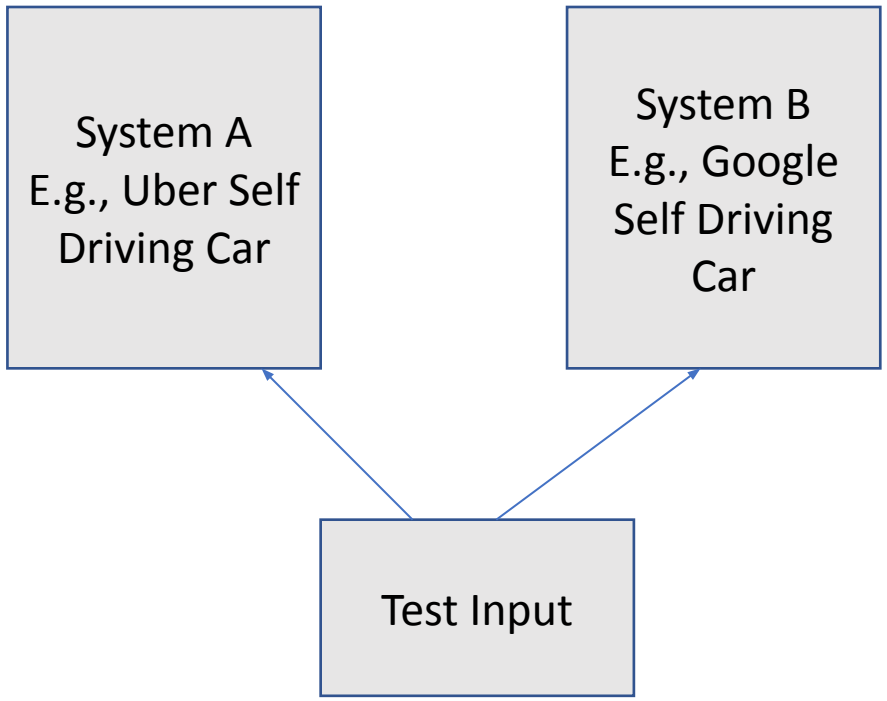
ii) Metamorphic Testing (MT)

Test Oracle and Oracle Problem

- Oracle is a mechanism where a program is verified by comparing its output with the expected outcome.



Differential Testing



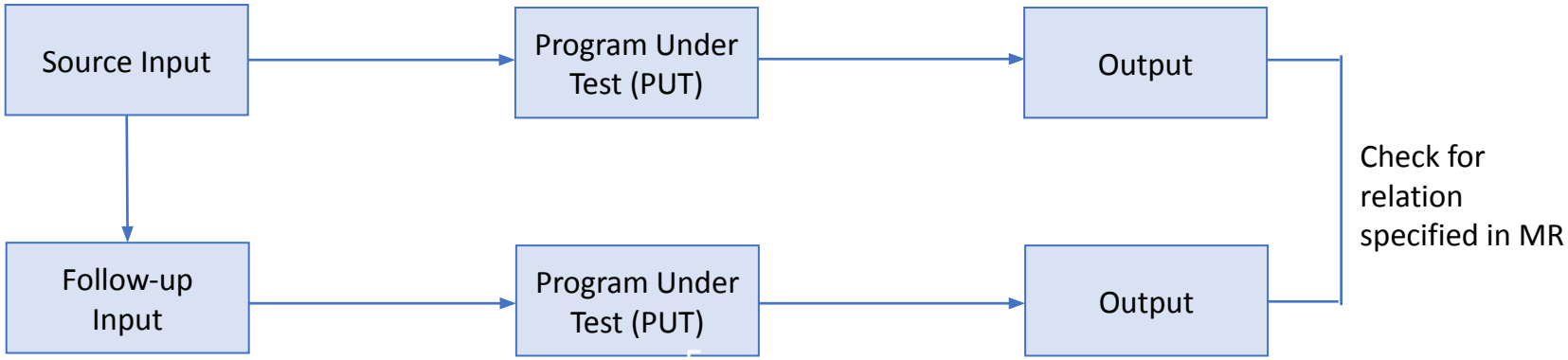
Metamorphic Testing (MT)

Metamorphic Testing uses Metamorphic Relations (MRs) to find defects in the program.

Metamorphic Relations (MRs) are the necessary properties of a program under test that specifies how the output should be changed on changing the related input.

Instead of verifying a single output, MT uses multiple execution to test the correctness of the program under test.

As an example, a program used to calculate the Standard deviation. What can be the MRs to test this program?



Metamorphic Testing for Software Quality Assessment: A Study of Search Engines [3]

- Popular search engines tested include Google, Bing, Chinese Bing, and Baidu

- As per research 65 percent of search clicks are done on the first result.

- **MR#1:** For example, let us test Google by issuing the following source query: “side effect of antibiotics in babies”. Let us consider the top result, which is: http://www.dailymail.co.uk/health/article2344802/Babies-given-antibiotics-theyreprone-eczema-Drugs-increase-risk-40.html.

- Even if the assessor is unable to verify or evaluate each individual response, he/she can still verify the logical consistency relationship among multiple responses.

- The metamorphic relation “MPSite” enables the construction of the following follow-up query: [“side effect of antibiotics in babies” site:uk], where “site:” is a Google search operator that specifies domains.

Metamorphic Testing for Software Quality Assessment: A Study of Search Engines [3]

- **MR#2 (Reverse the order):** The following is an example of the source query A: [“Newton” AND “Elvis” AND “Albert Einstein” AND “Plato”]. The follow-up query B is constructed by reversing the order of A’s terms: [“Plato” AND “Albert Einstein” AND “Elvis” AND “Newton.”] . The results should remain the same.

- **MR#3:** Adding a dot or space at the end should not change the results. This MR can be used to test the robustness of search engine under test.

MT for Cybersecurity [2]
Detecting Obfuscator Bugs

4 Obfuscators are tested: Cobfusc, Stunnix CXX-Obfus, Tigress, and Obfuscator-LLVM.

MR1 states that if two different source programs (P1 and P2) are functionally equivalent, their obfuscated versions (O(P1) and O(P2)) will also be functionally equivalent and, therefore, the compiled obfuscated executable programs, C(O(P1)) and C(O(P2)), should have equivalent behavior—the same outputs for the same inputs.

MR2 (Consistency): If the environment under which the obfuscator is run is treated as 'time' then for the same input program, the obfuscator should generate behaviorally equivalent programs regardless of when the obfuscator is run.

MR3 checks whether obfuscation rules have been applied consistently each time the obfuscator runs. Even if testers do not have detailed knowledge of the obfuscation rules, they can still check if outputs are consistent. For example, if a variable name in program P was obfuscated when the obfuscator ran yesterday, then the same variable name should still become obfuscated when the obfuscator is run today.

MT for Query-Based Systems [1]

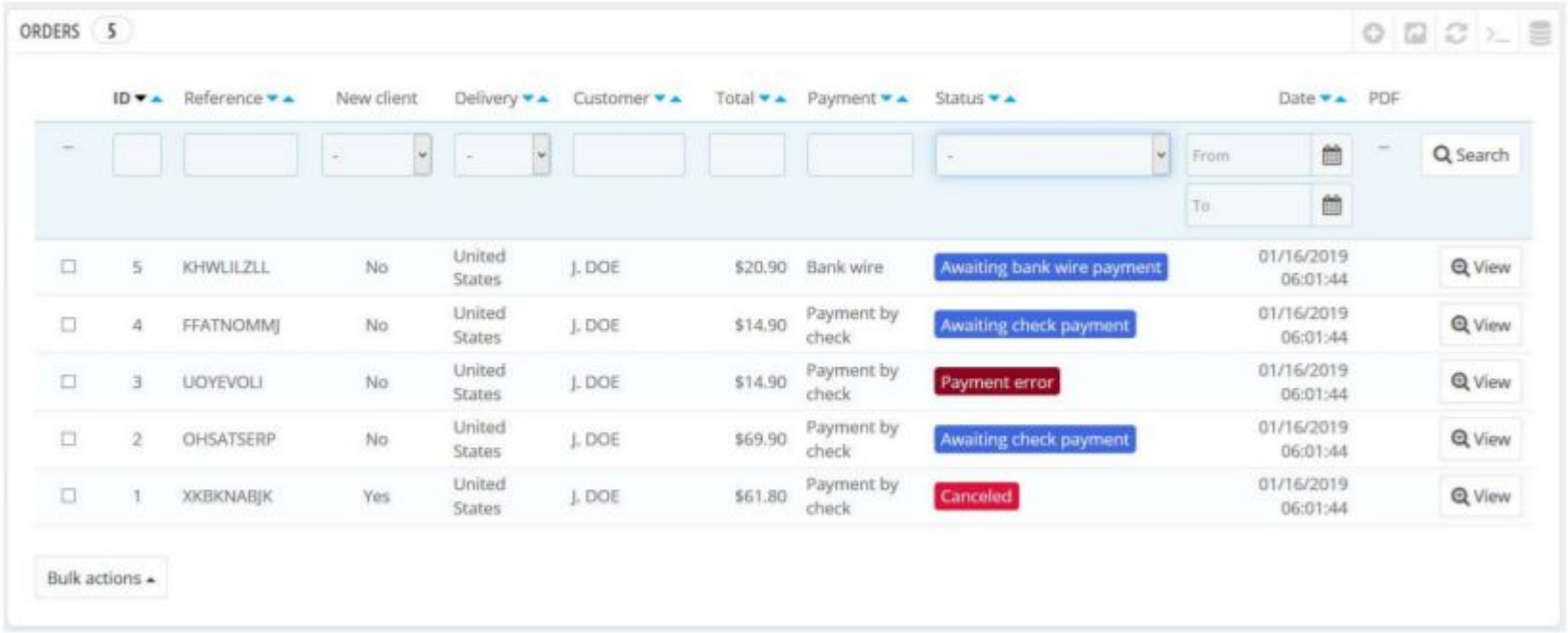


Figure 1. Order view in PrestaShop's back-office

MT for Query-Based Systems [1]

- *Identify the source and follow-up test cases?*

- **MR1 (Input equivalence):** List the orders using default values. Then, get a new list ordered by ID, default ordering criterion. Both queries should return exactly the same result set.

- **MR2 (Shuffling):** List the orders ordered by Date. Then, get a new list ordered by Total amount. Both queries should return the same orders, regardless of their ordering.

- **MR3:** List the orders placed after a date. Then, repeat the query adding a new filter to list orders paid by Bank wire only. The result set of the second query (follow-up test case) should be a subset of the result set of the first query (source test case), where no payment filter was applied.

MT for Query-Based Systems [1]

- **MR4 (Disjoint partitions):** List the orders with status Delivered. Then, repeat the query three more times changing the status filter to Cancelled, Refunded, and Payment error, respectively. The result sets of the four queries should have no orders in common.

- **MR5: Source Execution:** Search for orders from 1st Jan To 31 March.

  Follow-up: Search for orders from 1st Jan to 1st March, and then from 2nd March to 31st March, the results should remain the same for both the source and follow-up executions**.**

MT for Machine Translation (MT4MT) [4]

- In 2017, A Palestinian man was arrested by police after posting “Good morning” in Arabic which was wrongly translated as “attack them” by Facebook.

- Machine translators under test include Google and Microsoft translators.

MR: Round Trip Translation (RTT)

This MR states that first, translate a string L1 from the original language O to the target language T, resulting in a string L2; then, translate L2 back to the original language O, resulting in a string L3; finally, assess the translation quality by comparing L1 and L3: the closer, the better.

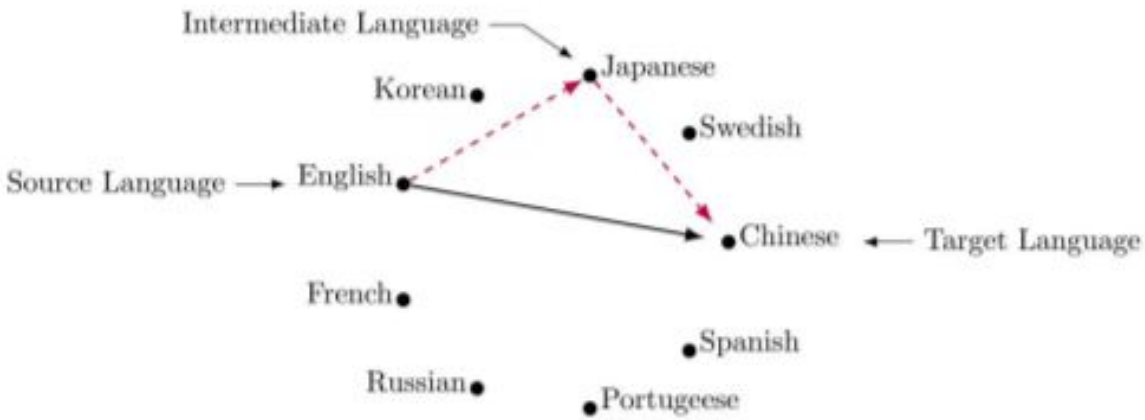
Problem / limitation of this MR?

MT for Machine Translation (MT4MT) [4]

- The main issue with RTT is that it involves two-way translations: the forward translation (FT) and the back translation (BT). Therefore, it could be hard to assess the quality of one-way translations

- **MR based on non-RTT:** The general idea was that a perfect translator should produce the same translation “when translating either directly (from a source language to a target language) or indirectly (from the source language to an intermediate language and then from the intermediate language to the target language)”.

MT for Machine Translation (MT4MT) [4]



MT for Machine Translation (MT4MT) [4]

MR_replace: A Helps B, C Helps B

- Subject-verb-object Structure

Fig. 4: An Example of translation inconsistency: The same "Nora" was translated into two different Chinese names by Microsoft Translator.

(a) Translation failure for "Layla."

(b) Translation failure for "Mini."

Fig. 5: Google Translate failures: The same names were sometimes translated and sometimes not translated.

MT for Image Recognition Systems

In the following MRs, what will be the source and follow-up test cases?

- **MR1:** Flipping the image should be recognized as original image.
- **MR2:** Adding a small noise i.e., adding a dot in the image should not change the output. It can be used for checking the robustness of a system.
- **MR3:** Rotation the image e.g., 25, 45 degrees should not change the output.
- **MR4:** Changing the color of object should not change the output.
- **MR5:** Slightly scaling an image should not change the output. It can be used for checking the robustness of a system.

Adversarial Testing ?

Addressing Data Quality Problems with Metamorphic Data Relations [5]

$$atr('username', U_0) \neq atr('username', U_1)$$

$$atr('foaf:birthDate', x_c) > atr('foaf:birthDate', x_p)$$

$$atr('dbo:father', child_0) = atr('dbo:father', child_1)$$

- Credit card utilization related data, can you propose MR?

MT For Test Case Generation

- MT is not only used as a testing strategy but can also be used for test case generation process.
- Let see, how MT can be used for test case generation purpose.

References

[1] Segura, S., Durán, A., Troya, J., & Ruiz-Cortés, A. (2019, May). Metamorphic relation patterns for query-based systems. In *2019 IEEE/ACM 4th International Workshop on Metamorphic Testing (MET)* (pp. 24-31). IEEE.

[2] T. Y. Chen *et al*., "Metamorphic Testing for Cybersecurity," in *Computer*, vol. 49, no. 6, pp. 48-55, June 2016, doi: 10.1109/MC.2016.176.

[3] Z. Q. Zhou, S. Xiang and T. Y. Chen, "Metamorphic Testing for Software Quality Assessment: A Study of Search Engines," in *IEEE Transactions on Software Engineering*, vol. 42, no. 3, pp. 264-284, 1 March 2016, doi: 10.1109/TSE.2015.2478001.

[4] L. Sun and Z. Q. Zhou, "Metamorphic Testing for Machine Translations: MT4MT," *2018 25th Australasian Software Engineering Conference (ASWEC)*, 2018, pp. 96-100, doi: 10.1109/ASWEC.2018.00021.

[5] F. Auer and M. Felderer, "Addressing Data Quality Problems with Metamorphic Data Relations," *2019 IEEE/ACM 4th International Workshop on Metamorphic Testing (MET)*, 2019, pp. 76-83, doi: 10.1109/MET.2019.00019.