Metamorphic Testing Part 1

Idaho State University | Computer Science

Isaac Griffith

CS 4422 and CS 5599
Department of Computer Science
Idaho State University

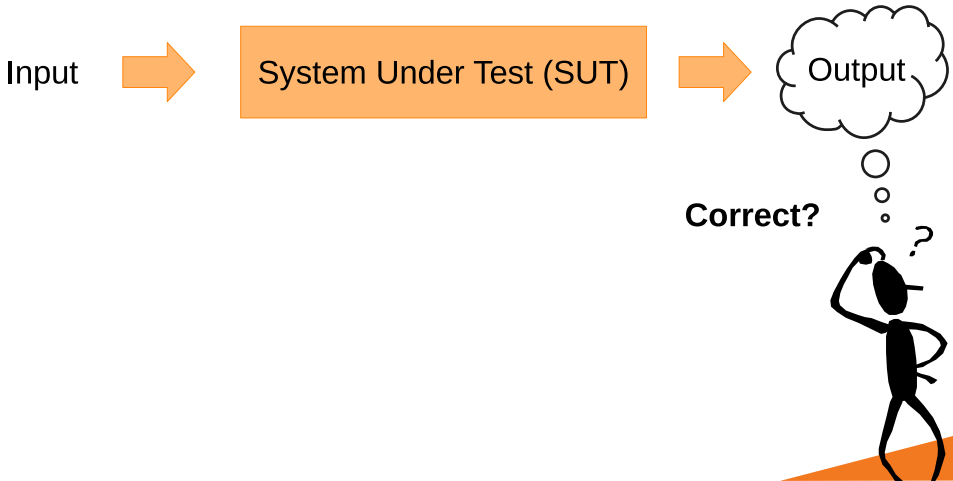
ROAR

Idaho State University
Computer Science

Introduction

Test Oracle

Mechanism to decide whether a test output is correct or not

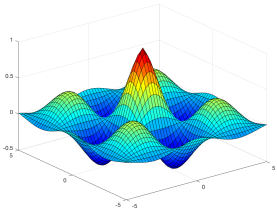


Idaho State University
Computer Science

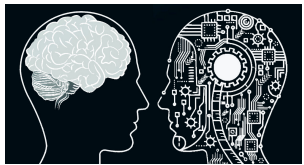
Introduction

Oracle Problem

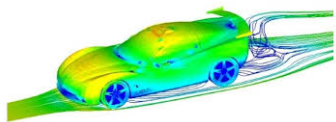
Sometimes it is not feasible to check the correctness of a test output



Scientific Calculations



Artificial Intelligence



Simulation & Modeling

ROAR

Let’s see some examples

Idaho State University
Computer Science

Examples

G

Source = s
Destination = d

shortestPath(G, s, d)

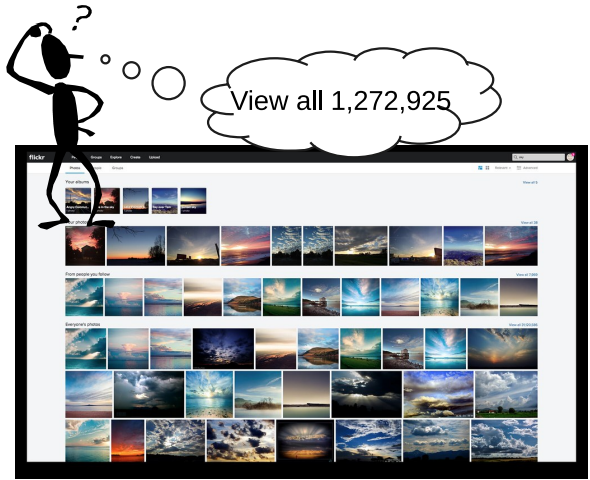
{e,g,h,t,x,z}

ROAR

Examples

flickr

Software



Idaho State University
Computer Science

Examples

Source test case

Graph G
Source s
Destination d

shortestPath(G, s, d)

{e, g, h, t, x, z}

|shortestPath(G, s, d)|
=
|shortestPath(G, d, s)|

Follow-up test case

Graph G
Source s
Destination d

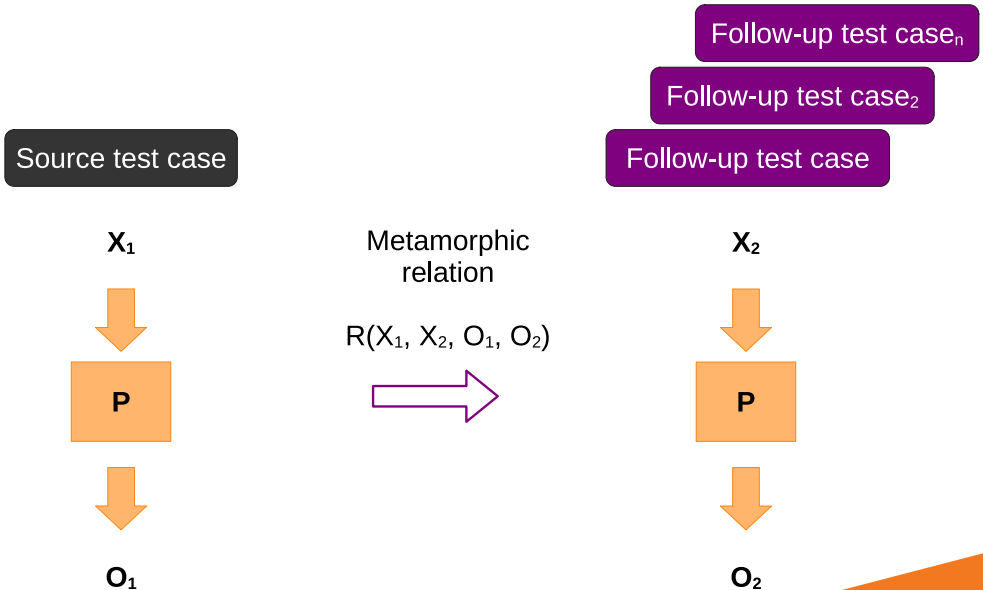
shortestPath(G, d, s)

{z, x, g, e}

ROAR

Idaho State University
Computer Science

Examples



ROAR

Idaho State University
Computer Science

Metamorphic Testing Process

1. Identification of metamorphic relations.
2. Generation/Selection of source test cases.
3. Generation of follow-up test cases.
4. Checking of metamorphic relations

ROAR

Oh, I get it. This is about alleviating the oracle problem. Is that it?

Yes! but MT can also support test data generation!

Idaho State University
Computer Science

Test Data Generation

|shortestPath(G, s, d)|
=
|shortestPath(G, d, s)|

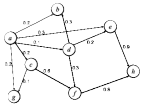
Metamorphic relation

Graph database

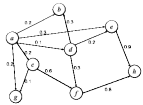
MT 1

|shortestPath(G, k, t)|
=
|shortestPath(G, t, k)|

MT 2



|shortestPath(P, c, a)|
=
|shortestPath(P, c, a)|



MT 3

|shortestPath(S, 2, 41)|
=
|shortestPath(S, 41, 2)|

ROAR

Idaho State University
Computer Science

Test Data Generation

If Q_2 == Q_1 AND size = large
then Count(Q_2) <= Count (Q_1)

Metamorphic relation

+

Random word generator

→

MT 1

Q_1 = “dog”
Q_2 = “dog”, size = large
Count(Q_2) <= Count(Q_1)

MT 2

Q_1 = “wind”
Q_2 = “wind”, size = large
Count(Q_2) <= Count(Q_1)

MT 3

Q_1 = “thing”
Q_2 = “thing”, size = large
Count(Q_2) <= Count(Q_1)

MT 4

Q_1 = “money”
Q_2 = “money”, size = large
Count(Q_2) <= Count(Q_1)

ROAR

State of the art

Domains

From a survey of 84 Case Studies

- Numerical programs ~5%
- Variability and decision support ~5%
- Compilers ~4%
- Components ~3%
- Autonomous Vehicles ~2%
- Bioinformatics ~8%
- Machine Learning ~8%
- Simulation and Modeling ~8%
- Embedded Systems ~8%
- Computer Graphics ~11%
- Web Services/Apps ~14%
- Other (Adobe, NASA, CyberSec) ~24%

Lessons Learned

Lesson Learned

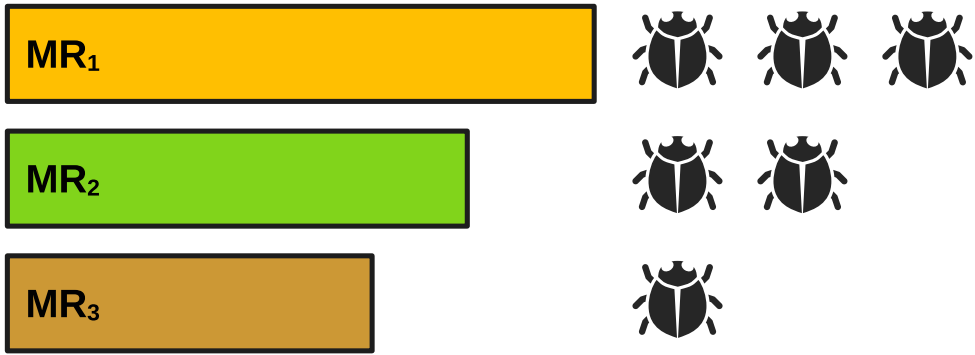
Metamorphic testing requires good knowledge of the problem domain

Idaho State University
Computer Science

Lessons Learned

Lesson Learned

Different metamorphic relations can have different fault-detection capability

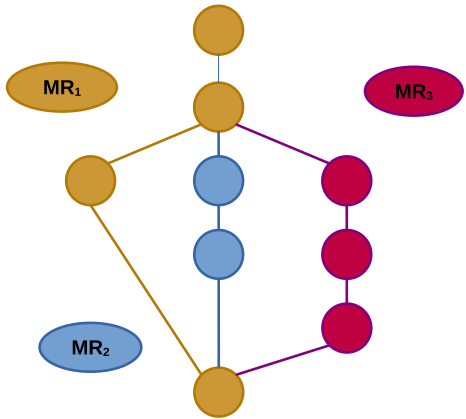


ROAR

Lessons Learned

Lesson Learned

Metamorphic relations should be diverse so they exercise different parts of the program.

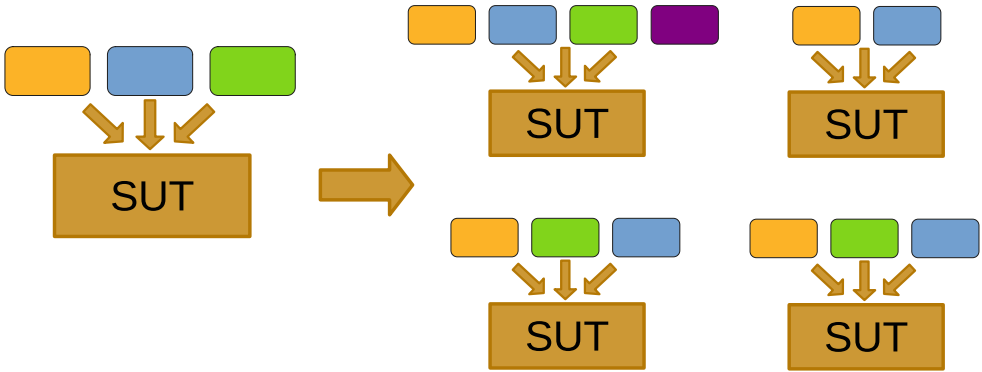


Idaho State University
Computer Science

Lessons Learned

Lesson Learned

Two common approaches for the construction of metamorphic relations: **input-driven** vs. output-driven



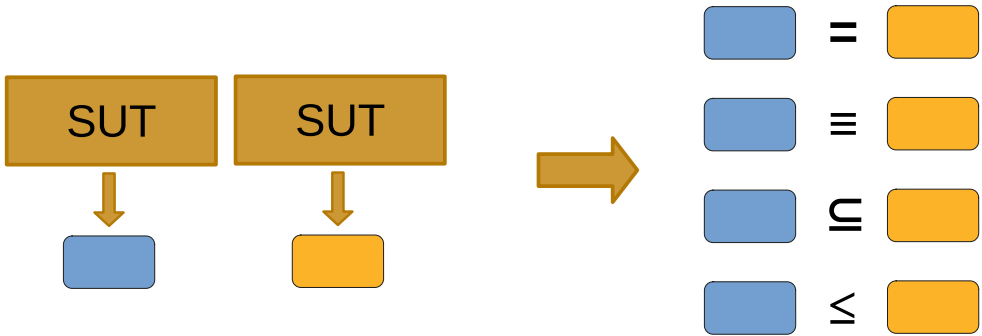
ROAR

Idaho State University
Computer Science

Lessons Learned

Lesson Learned

Two common approaches for the construction of metamorphic relations: input-driven vs. **output-driven**

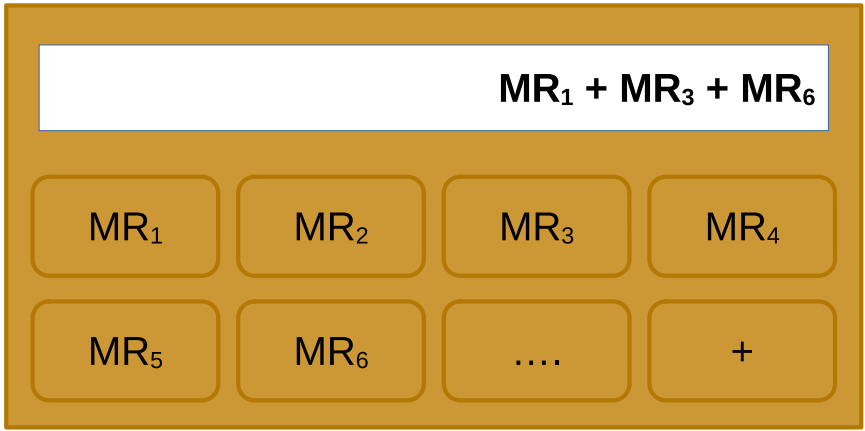


ROAR

Lessons Learned

Lesson Learned

Metamorphic relations can be combined

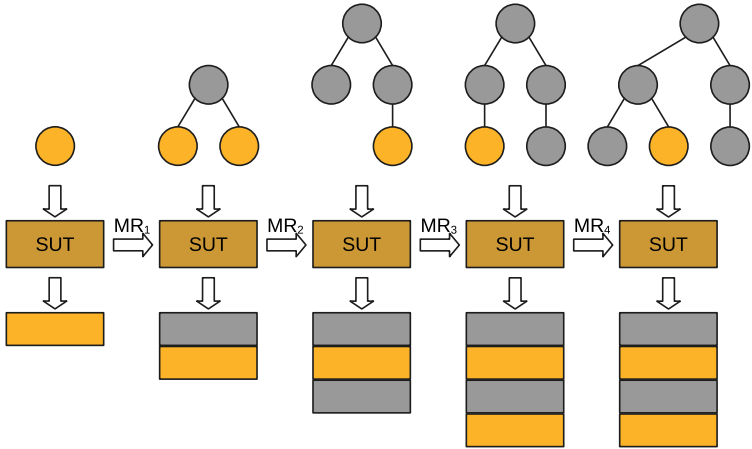


Idaho State University
Computer Science

Lessons Learned

Lesson Learned

Metamorphic relations can be combined



ROAR

Idaho State University
Computer Science

Lessons Learned

Lesson Learned

The automated discovery of metamorphic relations seems feasible in certain domains

ROAR

Challenges

- Systematic guidelines for the construction of good metamorphic relations
- Generation of likely metamorphic relations
- Non-functional metamorphic testing
- Provide tools to foster the use of the technique

Are there any questions?

Lecture borrowed from Segura and Zhou's presentation on "Metamorphic Testing 20 Years Later: A Hands-on Introduction" as presented at the 40th International Conference on Software Engineering.