CS301 Software Development

Instructor: Peter Kemper

An Oracle for Testing

Testing

- Definition

  “A test case is a set of inputs, execution conditions, and a pass/fail criterion”.

  ... plus some documentation ...

- A test case requires a pass/fail criterion:

  —> Oracle in testing literature.

- Delphi is not what is used to be:

Photo: flickr cc Konstantinos Kiriazopoulos
Hellas Delphi Oracle

The Oracle Problem

- The key question:

How do we decide if the observed program behavior is correct?

- Very often text books dance around this problem:
  - Correct solutions are trivial:

    Length of a queue with 3 entries is 3.
  - Manual calculations to obtain a solution are feasible or known: 2 + 2 = 4
  - Observation of failure trivial: SUT crashes

Observation 1: We pick the question!

- Some questions may be easier to answer than others.
- Possible sources of information
  - Documentation, Specification
  - Published results, reference values
  - Subject matter expert

- For certain questions, answer feasible
  - with manual calculations:
    - Values such as 10,000 as good as 17,845
    - Constants can be 0, coefficients can be 1
  - with analytical formulas that are known and tractable
    - exact answer, or lower/upper bound

Observation 2: Redundancy in Algorithms

- Often many algorithms known to solve a problem

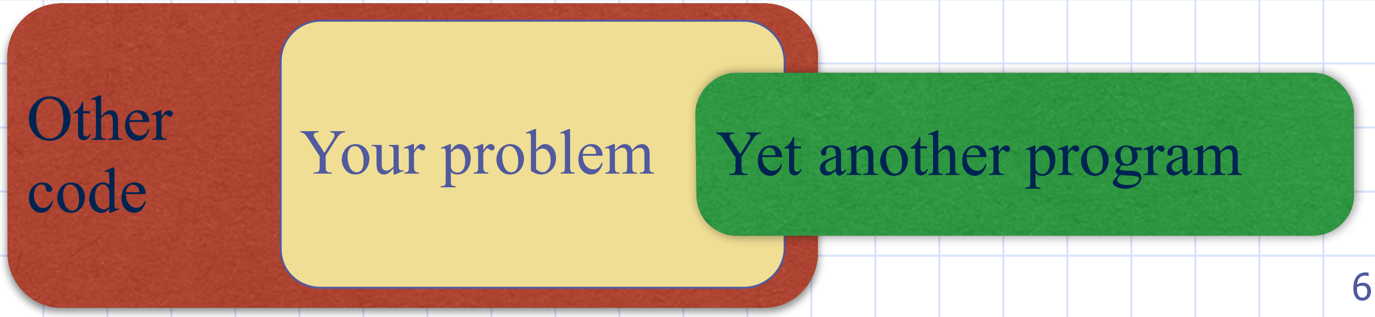
- Start with a “naïve” version that is straightforward
  - Little effort to implement
  - Less effort to evaluate
  - Low risk for mistakes

- Implement algorithm you want to use in production code

- Use simple algorithm as oracle for production version
  - Basically check for consistency

Observation 3: There is a lot of code out there!

- Use existing code as an Oracle
- Examples
  - Use other dedicated frameworks you cannot use in production code, e.g. matlab, mathematica
    - Use something much more general
    - Use something much more special and limited
  - Use a different programming environment that is useful
    - e.g., scripting may be more productive but not necessarily efficient, platform independent, portable, ..
  - Use inefficient but readily available data structures
  - Use inefficient brute force naïve algorithms from a library



Other code

Your problem

Yet another program

Observation 4: Consistency

- Consistency is weaker than correctness but still useful

- Often solution to one task helps to find solution for other
  - Relations: reflexive, symmetric, transitive, …
  - Algebraic: associate, commutate, distributive, …
  - Geometric: reflection, rotation
  - Permutation of parameter values may lead same answer or inverse answer:
    - compare(A,B) vs compare(B,A)

- Compute solution to a set of related problems and check if answers are consistent
  - Risk: answers are all wrong but consistent

Summary on Oracle Problem

- Oracle requires some kind of redundant solution for specific test questions
  - Manual solution for simple cases
  - Subject matter expert
  - Specification, documentation
  - 2nd implementation
    - own implementation of simple algorithm, alternative code for different environment etc.
    - external existing code (more general or more specialized)

- Outcome of Oracle needs to be documented in test case
  - if code fails test, prime suspect of failure is test case not code

- Test code must be simple enough such that it need no testing on its own (vicious circle)

and again

- There should be no magic involved in this!
- Transparency is important for credibility
  - How did you get the correct answer? Why is it correct?

Photo: flickr cc Konstantinos Kiriazopoulos
Hellas Delphi Oracle

Photo: flickr cc Xuan Che
Oracle Bones, Ancient China