Test Oracles and Randomness

Ralph Guderlei and Johannes Mayer

rjg@mathematik.uni-ulm.de, jmayer@mathematik.uni-ulm.de

University of Ulm, Department of Stochastics and Department of Applied Information Processing

Introduction

- Standard methods of software testing do not provide information about software reliability.
  $\Rightarrow$ random testing
- The main problem in random testing is to verify the actual results of the Implementation Under Test (IUT).
  $\Rightarrow$ Oracles

Oracles

An oracle provides methods

- to generate expected results for test cases
- to compare the expected results to the actual results of the IUT.

Oracles

An oracle provides methods

- to generate expected results for test cases
- to compare the expected results to the actual results of the IUT.

It consists of two parts:

- the result generator to obtain expected results
- the comparator to verify the actual results of the IUT

Standard Types of Oracles

Oracles do not apply generally, only in special cases.

Standard types are

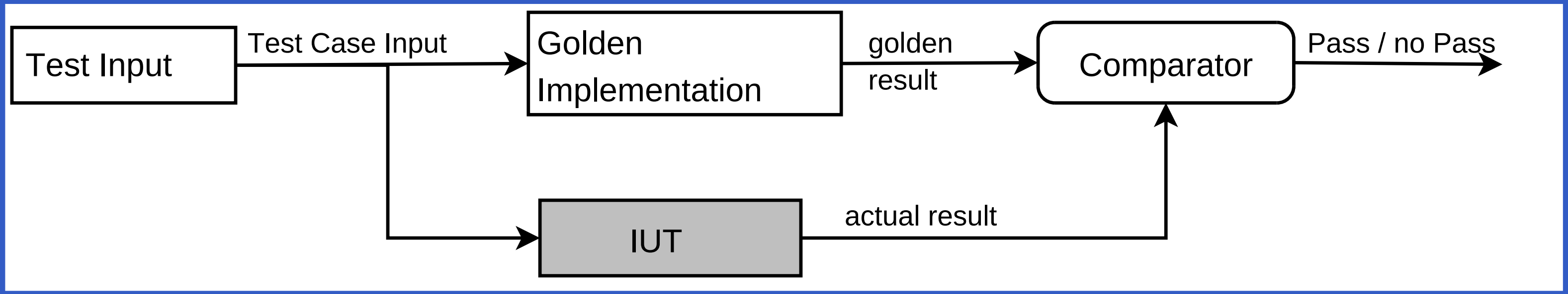
- Perfect Oracle
- Gold Standard Oracle
- Parametric Oracle or Heuristic Oracle

Perfect Oracle

- equivalent to the IUT and completely trusted
- accepts every input specified for the IUT
- produces ”always” the correct result

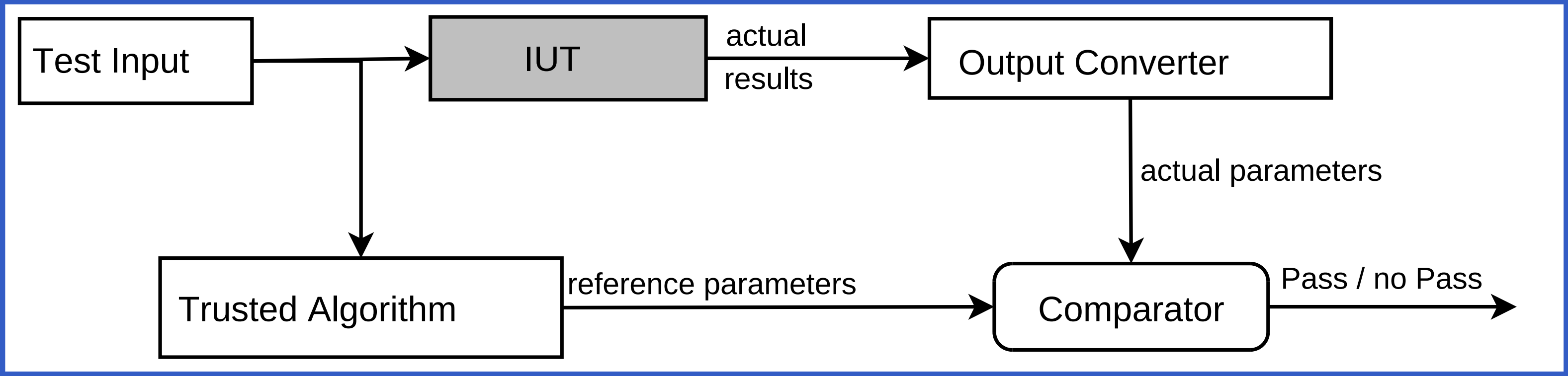
$\Rightarrow$ a ”defect free” version of the IUT

Gold Standard Oracle



Use one or more versions of an existing application system to generate expected results (e.g. a legacy system).

Parametric Oracle



Use an algorithm to compute parameters from the actual results and compare the actual parameters to expected parameter values.

Statistical Oracle

- special case of a parametric oracle

Statistical Oracle

- special case of a parametric oracle
- parameters are computed with statistical tools

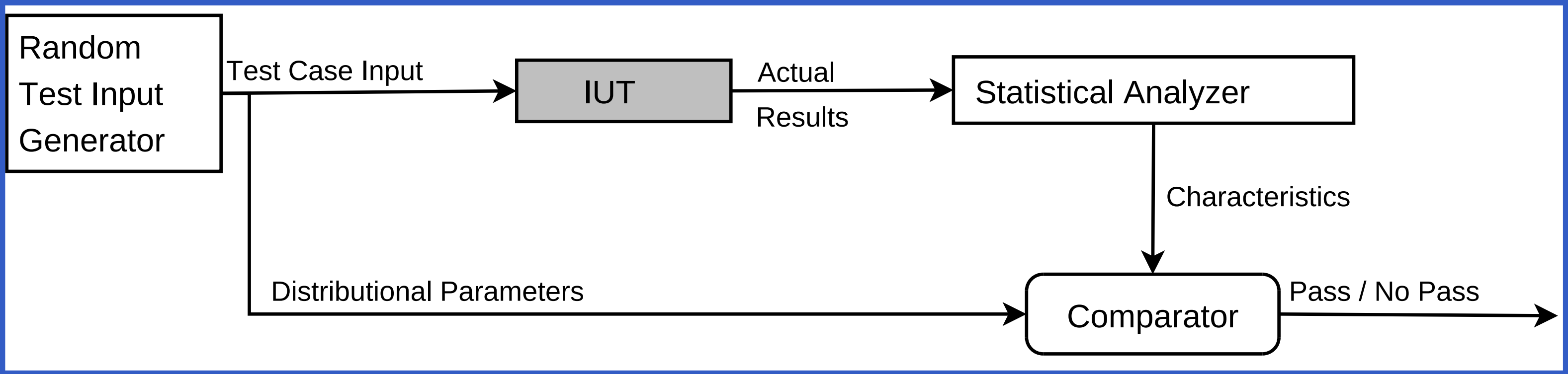
Statistical Oracle

- special case of a parametric oracle
- parameters are computed with statistical tools
- comparison is done in a statistical way

Statistical Oracle

- special case of a parametric oracle
- parameters are computed with statistical tools
- comparison is done in a statistical way
- generation of random input data allows a large number of test cases

Micropattern



distributional properties of the generator and the IUT are used for the comparison

Requirements

- statistical characteristics of the IUT have to be known
- large number of input data for stable results (> 30)

Possible Uses

- (Scientific) applications dealing with randomness e.g. simulators, data analysis (e.g. in banking, image analysis)
- Applications with complicated input data, where reference values are difficult to obtain. As in the Example:
    - images are difficult to analyze
    - randomly generating and comparing the mean values is simple

Necessary Statistics

- X_i iid random variables with mean μ and variance σ^2

Necessary Statistics

- X_i iid random variables with mean μ and variance σ^2
- sample mean of n random variables:
  $$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Necessary Statistics

- X_i iid random variables with mean μ and variance σ^2
- sample mean of n random variables:

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

- sample variance of n random variables:

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2$$

Distributional Properties

using the central limit theorem, one can assume the X_i to be (assymptotically) normally distributed.

$$\sqrt{n}\frac{\bar{X}_n - \mu}{\sigma} \xrightarrow{d} \mathcal{N}(0,1) \text{ as } n \to \infty$$

Simple Approach

- expected value is known: μ_0

Simple Approach

- expected value is known: μ_0
- compute empirical mean of the n actual results x_i:
  $\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$

Simple Approach

- expected value is known: μ_0
- compute empirical mean of the n actual results x_i:
  $\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i$
- pass if $\frac{|\bar{x}_n - \mu_0|}{\mu_0} < \varepsilon$, e.g.$\varepsilon = 0.1$

First Attempt

Natural choice: t-test if the mean of the actual results μ is equal to the expected result μ_0.

First Attempt

Natural choice: t-test if the mean of the actual results μ is equal to the expected result μ_0.
Type I error: IUT does not pass though it is correct (false alarm)
Type II error: IUT does pass though it is not correct (false pass)

But **only** the Type I error will be controlled.

Advanced Approach

Alternative choice: use the *intersection-union method* to invert the test hypothesis.
So the controllable probability for the Type I error becomes the probability for a false pass.

Advanced Approach

Alternative choice: use the *intersection-union method* to invert the test hypothesis.
So the controllable probability for the Type I error becomes the probability for a false pass.

With $\delta > 0$, one can define an interval around μ_0 such that

$$\bar{X}_n \notin [\mu_0 - \delta, \mu_0 + \delta]$$

for a given probability α.

The Test Statistic

For a given probability $\alpha \in (0, \frac{1}{2})$, the IUT passes if

$$\sqrt{n}\frac{\bar{x}_n-(\mu_0-\delta)}{s_n} \geq t_{n-1,\frac{\alpha}{2}}$$

and

$$\sqrt{n}\frac{\bar{x}_n-(\mu_0+\delta)}{s_n} \leq t_{n-1,\frac{\alpha}{2}},$$

The Test Statistic

For a given probability $\alpha \in (0, \frac{1}{2})$,
the IUT passes if

$$\sqrt{n}\frac{\bar{x}_n-(\mu_0-\delta)}{s_n} \geq t_{n-1,\frac{\alpha}{2}}$$

and

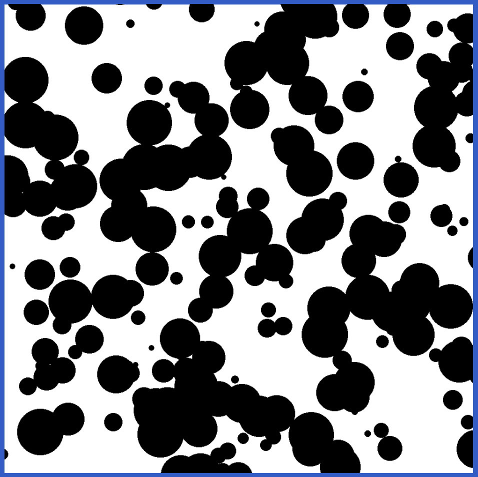
$$\sqrt{n}\frac{\bar{x}_n-(\mu_0+\delta)}{s_n} \leq t_{n-1,\frac{\alpha}{2}},$$

where $t_{n,\frac{\alpha}{2}}$ denotes the $(1-\alpha/2)$ - quantile of the Student t-distribution with $n-1$ degrees of freedom.

An Example from Image Analysis

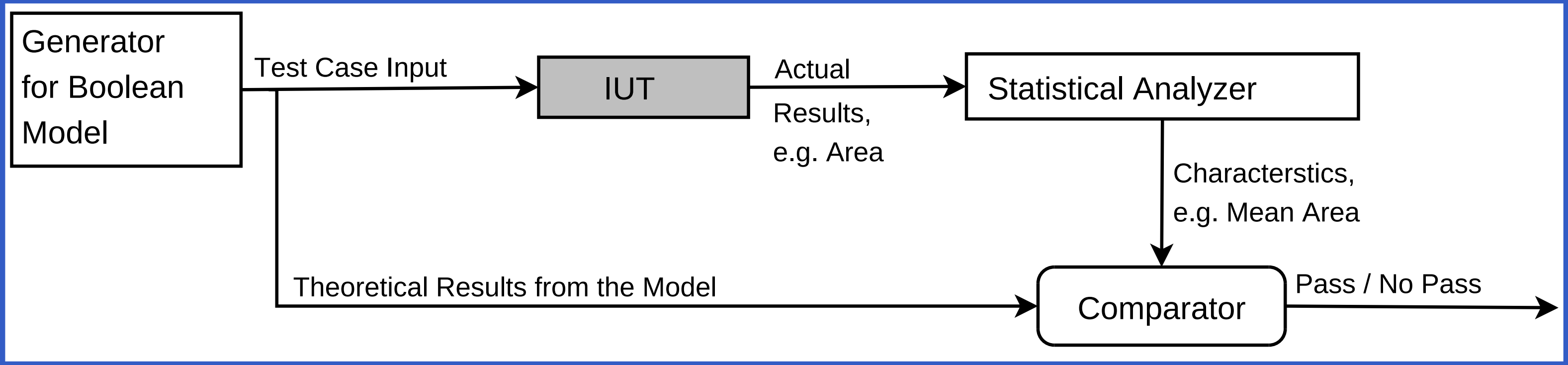
- compute morphological properties such as area or boundary length
- fit stochastic models to the given data

Random Input - The Boolean Model



- computationally simple model
- flexible
- good fit to real data
- the expected mean area and mean boundary length are known in explicit form

Micropattern Revisited



- Random Input Generator = Generator for the Boolean Model
- The IUT computes e.g. the area or the boundary length

Usage of the Simple Approach

Performing the test

- for each computed characteristic (e.g. the area)
- for different Boolean Models

Usage of the Simple Approach

Performing the test

- for each computed characteristic (e.g. the area)
- for different Boolean Models

The simple approach was used as

- advanced smoke test to detect severe bugs in the program flow
- plausibility check for the computed values

Usage of the Advanced Approach

- verification of the results of the simple approach
- choosing a small α, one can say that the IUT produces the correct results
  - (only) with respect to the tested characteristics
  - with probability α

Numerical Example

- sample of $n = 500$ images, error probability $\alpha = 0.0001$, here $x_i \hat{=}$ boundary length

Numerical Example

- sample of $n = 500$ images, error probability $\alpha = 0.0001$, here $x_i \hat{=}$ boundary length
- $\bar{x}_n = 0.044925685$, $\mu_0 = 0.044948$, $\delta = 0.00044948 = 1\% * \mu_0$

Numerical Example

- sample of $n = 500$ images, error probability $\alpha = 0.0001$, here $x_i \hat{=}$ boundary length
- $\bar{x}_n = 0.044925685$, $\mu_0 = 0.044948$, $\delta = 0.00044948 = 1\% * \mu_0$
- $\sqrt{n}\frac{\bar{x}_n - (\mu_0 - \delta)}{s_n} = 10.38 > 3.92 = t_{499,0.99995}$ and

$$\sqrt{n}\frac{\bar{x}_n - (\mu_0 + \delta)}{s_n} = -11.47 < -3.92 = -t_{499,0.99995}.$$

Numerical Example

- sample of $n = 500$ images, error probability $\alpha = 0.0001$, here $x_i \hat{=}$ boundary length
- $\bar{x}_n = 0.044925685$, $\mu_0 = 0.044948$, $\delta = 0.00044948 = 1\% * \mu_0$
- $\sqrt{n}\frac{\bar{x}_n-(\mu_0-\delta)}{s_n} = 10.38 > 3.92 = t_{499,0.99995}$ and

$$\sqrt{n}\frac{\bar{x}_n - (\mu_0 + \delta)}{s_n} = -11.47 < -3.92 = -t_{499,0.99995}.$$

- So the IUT passes the test.

Conclusion

- The approach includes information about the reliability with respect to for the tested characteristics
- The approach makes it possible to handle randomness in tests
- It is possible to test for other characteristics than the mean
- The approach shown above does not apply in all cases

Thank you for your attention.