Even the Very Wise Cannot See All Ends:
Many Facets of the Test Oracle Problem

Prof. T.H. Tse
The University of Hong Kong
Pokfulam, Hong Kong
Formula for
A Successful Keynote Speech

{ ?!" " _ ^ \notin \rightarrow :-) }
Formula for
A Successful Keynote Speech

\{ ? ! " " _ ^ \notin \rightarrow :-) \}
Formula for A Successful Keynote Speech

{ ?"" "_^不属于 -> :-) }

Comprehensive coverage

Bold questions
Formula for A Successful Keynote Speech

{ ?! " " _ ^ ∉ -> :- ) }

- Comprehensive coverage
- Bold questions
- Provocative assertions
Formula for A Successful Keynote Speech

{ ?!" " _ ^ \notin \rightarrow :-) }
Solid foundations
Formula for A Successful Keynote Speech

\{ ? ! " " _ ^ \notin - > :) \}
Formula for A Successful Keynote Speech

{ ?!" " _ ^ ∉ - > : - ) }
Formula for A Successful Keynote Speech

\{ ?!" " _ ^ \notin \to :-) \}
Formula for A Successful Keynote Speech

{ ?!" " _ ^ \notin -> :-) }

Interesting stories

Real-world applications
Formula for A Successful Keynote Speech

{ ? ! " " _ ^ \notin \rightarrow :-) }

- Inspiring conclusions.
- Interesting stories
- Real-world applications
Presentation Outline

- Formula for a successful keynote speech
- A successful keynote speech.
Presentation Outline

- Background
Presentation Outline

- Background
- Many facets of the test oracle problem
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome
    = actual execution result
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
- Jungle of proposals
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result

- Jungle of proposals
- Empirical studies?
Presentation Outline

◆ Background
◆ Many facets of the test oracle problem
  ▪ Expected outcome = actual execution result
  ▪ Expected outcome = actual execution result
  ▪ Expected outcome = actual execution result

◆ Jungle of proposals
◆ Empirical studies?
◆ What do other researchers do?
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result

- Jungle of proposals
- Empirical studies?
- What do other researchers do?
- Trim the tree or tame the forest?
Presentation Outline

- **Background**
  - Many facets of the test oracle problem
    - Expected outcome \(=\) actual execution result
    - Expected outcome \(=\) actual execution result
    - Expected outcome \(=\) actual execution result

- Jungle of proposals
- Empirical studies?
- What do other researchers do?
- Trim the tree or tame the forest?
Summary of My Work

- Testing of object-oriented software
  - Black and White \([ACM TOSEM 1998]\)
  - TACCLE \([ACM TOSEM 2001]\)
  - VITAMIN \([Communications of the ACM 2007]\)
  - Equality to Equals and Unequals \([IEEE TSE 2013]\).
Summary of My Work

- **Testing pervasive software**
  - [COMPSAC 2004 best paper]
  - [FSE 2006]
  - [ICSE 2008].
Summary of My Work

- **Testing services computing**
  - [ICSE 2008b]
  - [FSE 2009]
  - [WWW 2009]
  - [IEEE TSC 2015 spotlight paper]
  - [IEEE TSC 2015b].
Summary of My Work

- **Testing based on formal specifications**
  - Tabular Expressions [*IEEE TSE 2011*]
Summary of My Work

- **Testing based on formal specifications**
  - Tabular Expressions [*IEEE TSE 2011*]

- **Testing based on informal specifications**
  - CHOC’LATE [*IEEE TSE 2003*]
  - [Communications of the ACM 2010]
  - DESSERT [*IEEE TSE 2012*].
Summary of My Work

- Spectrum-based fault localization
  - [COMPSAC 2008 best paper]
  - [FSE 2009b]
  - [IEEE Computer 2012]
Summary of My Work

- **Spectrum-based fault localization**
  - *COMPSAC 2008 best paper]*
  - [*FSE 2009b]*
  - [*IEEE Computer 2012]*

- **Debugging of concurrent systems**
  - [*Information Sciences 2012]*
  - [*ISSTA 2012]*.
Summary of My Work

- Integration of testing, debugging, and proving
  - [COMPSAC 2009 best paper]
  - [QSIC 2011 best paper]
  - [IEEE TSE 2011b].
Selected PhD Graduates

Dr W.K. Chan, Associate Professor, City University of Hong Kong

Dr Zhenyu Zhang, Associate Professor, Institute of Software, Chinese Academy of Sciences

Dr Bo Jiang, Associate Professor, Beihang University

Dr Lijun Mei, IBM Research — China

Dr Ke Zhai, Goldman Sachs.
Selected PhD Graduates

Dr W.K. Chan, Associate Professor, City University of Hong Kong

Dr Yan Cai, Associate Professor, Institute of Software, Chinese Academy of Sciences.

Dr Zhenyu Zhang, Associate Professor, Institute of Software, Chinese Academy of Sciences

Dr Bo Jiang, Associate Professor, Beihang University

Dr Lijun Mei, IBM Research — China

Dr Ke Zhai, Goldman Sachs
Presentation Outline

◆ Background

◆ Many facets of the test oracle problem
  ■ Expected outcome
    = actual execution result
  ■ Expected outcome
    = actual execution result
  ■ Expected outcome
    = actual execution result

◆ Jungle of proposals
◆ Empirical studies?
◆ What do other researchers do?
◆ Trim the tree or tame the forest?
The Test Oracle Problem

Even the Very Wise Cannot See All Ends:
Many Facets of the Test Oracle Problem

Prof. T.H. Tse
The University of Hong Kong
Pokfulam, Hong Kong

Present 20 years of work in 45 minutes
Many Facets of the Test Oracle Problem

Even the Very Wise Cannot See All Ends

J.R.R. Tolkien
The Lord of the Rings

Prof. T.H. Tse
The University of Hong Kong
Pokfulam, Hong Kong
Many Facets of the Test Oracle Problem

Even the Very Wise Cannot See All Ends

Many Facets of the Test Oracle Problem

Prof. T.H. Tse
The University of Hong Kong
Pokfulam, Hong Kong

J.R.R. Tolkien
The Lord of the Rings

- Rawlinson and Bosworth Professor of Anglo-Saxon, University of Oxford (1925–1945)
- Merton Professor of English Language and Literature, University of Oxford (1945–1959).
Oracle

- A message supposedly from God’s inspiration, usually given by a prophet
Oracle

- A message supposedly from God’s inspiration, usually given by a prophet
- An utterance of deep import or wisdom; an opinion or declaration regarded as authoritative and infallible; undeniable truth.
Oracle

- A message supposedly from God’s inspiration, usually given by a prophet
- An utterance of deep import or wisdom; an opinion or declaration regarded as authoritative and infallible; undeniable truth.
Test Oracle

- A test oracle is a mechanism to check whether
  - expected outcome according to the specification
    = actual result of executing the implementation.
Real-Life Search Engine Example
Real-Life Search Engine Example

Is 2.77M results correct?
Many Facets of the Test Oracle Problem

Challenge 1:

- **Expected outcome** = actual execution result
Metamorphic Testing

\[ \sin 0.9876 \]  

Expected \[ ? \]  
Actual \[ 0.8347 \]
Metamorphic Testing

- Cannot be verified because we do not know what to expect
## Metamorphic Testing

<table>
<thead>
<tr>
<th>sin 0.9876</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>0.8347</td>
</tr>
</tbody>
</table>

- Cannot be verified because we do not know what to expect
- Take a follow-up test case:

<table>
<thead>
<tr>
<th>sin ((\pi - 0.9876))</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>?</td>
<td>0.8347</td>
</tr>
</tbody>
</table>
Metamorphic Testing

\[
\sin 0.9876
\]

\[
\sin (\pi - 0.9876)
\]

Expected ? Actual 0.8347

Expected ? Actual 0.8347
<table>
<thead>
<tr>
<th>Expression</th>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sin 0.9876 )</td>
<td>?</td>
<td>0.8347</td>
</tr>
<tr>
<td>( \sin (\pi - 0.9876) )</td>
<td>?</td>
<td>0.8347</td>
</tr>
</tbody>
</table>
Metamorphic Testing

\[
\sin(0.9876) = \sin(\pi - 0.9876)
\]

Expected metamorphic relation

Expected: 
Actual: 0.8347

Expected: 
Actual: 0.8347
Expected metamorphic relation

\[ \sin 0.9876 = \sin (\pi - 0.9876) \]
Metamorphic Testing

Expected metamorphic relation

\[ \sin 0.9876 = \sin (\pi - 0.9876) \]

Expected

Actual

0.8347
Real-Life Search Engine Example
Apply Metamorphic Testing

Real-Life Example

Metamorphic Relation:
More refined search should produce fewer number of entries.
Real-Life Search Engine

Refine to “children’s hospital of chicago”.

More entries indicate failure in search engine.
Real-Life Search Engine Example

Is 4.54M results correct?.
Real-Life Search Engine

Refine to “children’s hospital of chicago”.

More entries indicate failure in search engine.
Real-Life Search Engine Example

Human Issues

😊 Microsoft Research selected our project for a Virtual Earth award
😊 Each of my brothers has 2 doctoral degrees
😊 I have only one doctoral degree.

😞 Google asked us to submit a bug report
Metamorphic Testing

Other Real-World Applications

- Services computing
- Ubiquitous computing
- Concurrent systems
- Graphic applications
- Numerical programs.
Based on metamorphic testing, an “equivalence modulo inputs” technique identified 147 faults in GCC and LLVM using one metamorphic relation.
Metamorphic Testing

Recent Track Records

Compilers

- Based on metamorphic testing, an “equivalence modulo inputs” technique identified 147 faults in GCC and LLVM using one metamorphic relation

Siemens Suite

- Detected 3 new faults in Siemens suite after its long history of test case studies.
Many Facets of the Test Oracle Problem

Challenge 2:

- Expected outcome $= \text{actual execution result}$
Many Facets of the Test Oracle Problem

Challenge 2:

- Expected outcome $\equiv$ actual execution result
- Consider the testing of real-life object-oriented software.
Technology-transfer project for ASM, the world’s largest supplier of assembly and packaging equipment for the semiconductor industry.
Testing of Object-Oriented Software

Robotic arm after handling *problem* chip

Simple concept in specification

Complex implementation

Expected

Home afresh

Actual

Home afresh
Many Facets of the Test Oracle Problem

Challenge 2:
- Expected outcome \( = \) actual execution result
- Consider the testing of real-life object-oriented software

Cannot define “=” at two different levels of abstraction.
### Testing of Object-Oriented Software

#### Mimic Metamorphic Testing?

<table>
<thead>
<tr>
<th>Robotic arm after handling <em>problem</em> chip</th>
<th>Expected Home afresh</th>
<th>Actual Home afresh</th>
</tr>
</thead>
</table>
Mimic Metamorphic Testing?

Robotic arm after handling problem chip

Robotic arm after handling standard chip
## Testing of Object-Oriented Software

### Mimic Metamorphic Testing?

<table>
<thead>
<tr>
<th>Robotic arm after handling <em>problem</em> chip</th>
<th>Expected</th>
<th>Home afresh</th>
<th>Actual</th>
<th>Home afresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotic arm after handling <em>standard</em> chip</td>
<td>Expected</td>
<td>Home afresh</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Testing of Object-Oriented Software

**Mimic Metamorphic Testing?**

<table>
<thead>
<tr>
<th>Robotic arm after handling <em>problem</em> chip</th>
<th>Expected</th>
<th>Home afresh</th>
<th>Actual</th>
<th>Home afresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotic arm after handling <em>standard</em> chip</td>
<td>Expected</td>
<td>Home afresh</td>
<td>Actual</td>
<td>Home afresh</td>
</tr>
</tbody>
</table>
### Testing of Object-Oriented Software

**Mimic Metamorphic Testing?**

<table>
<thead>
<tr>
<th>Robotic arm after handling <em>problem</em> chip</th>
<th>Expected</th>
<th>Home afresh</th>
<th>Actual</th>
<th>Home afresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotic arm after handling <em>standard</em> chip</td>
<td>Expected</td>
<td>Home afresh</td>
<td>Actual</td>
<td>Home afresh</td>
</tr>
</tbody>
</table>
Testing of Object-Oriented Software

Mimic Metamorphic Testing?

Robotic arm after handling *problem* chip

Robotic arm after handling *standard* chip

Expected

Home afresh

Actual

Home afresh

Expected

Home afresh

Actual

Home afresh
Testing of Object-Oriented Software
Mimic Metamorphic Testing?

Robotic arm after handling *problem* chip

Robotic arm after handling *standard* chip

Expected relation

Equivalent
Robotic arm after handling *problem* chip

Robotic arm after handling *standard* chip

Expected relation

Equivalent
Testing of Object-Oriented Software

Mimic Metamorphic Testing?

Robotic arm after handling *problem* chip

Robotic arm after handling *standard* chip

Expected relation

*Equivalent*

Actual relation

*Equivalent*
Target of OO Software Testing

- An implementation $P$ is correct with respect to the specification $Sp$ if and only if
  - For any pair of equivalent sequences of operations in $Sp$, the actual objects resulting from $P$ must be equivalent.
Challenge 3:

- Object A $\equiv$ object B

What is object equivalence?
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security. He was on the search committee for the editor-in-chief of IEEE Transactions on Software Engineering in 2013.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Associate Professor in Computer Science at The Chinese University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security. He was on the search committee for the editor-in-chief of IEEE Transactions on Software Engineering in 2013.

Click “Hidden”.
What is Object Equivalence?

Real-Life Word Processing Example

Call this object $H$.

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Output from Object $H$

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security. He was on the search committee for the editor-in-chief of *IEEE Transactions on Software Engineering* in 2013.

Take object $O$ again.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security. He was on the search committee for the editor-in-chief of *IEEE Transactions on Software Engineering* in 2013.
What is Object Equivalence?

Real-Life Word Processing Example

Call this object $D$.

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Output from Object $D$

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Attributive Equivalence

Two objects will be *attributively equivalent* if they have the exactly the same *visible attributes*.
What is Object Equivalence?

Attributive Equivalence

- Simple to test
- **But** the definition is too weak to be useful ...
- Why? 

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

Take object $H$ again (with hidden text).

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong and a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

Click “¶”.

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Associate Professor in Computer Science at The Chinese University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security. He was on the search committee for the editor-in-chief of IEEE Transactions on Software Engineering in 2013.
What is Object Equivalence?

Real-Life Word Processing Example

We get an object that behaves like the original object $O$.

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong—a Steering Committee Chair of the IEEE International Conference on Software Reliability & Security. He was on the selection committee for the editor-in-chief of *IEEE Transactions on Software Engineering* in 2013.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong, Steering Committee Chair of the IFIP International Conference on Software Quality, Reliability & Security.

Take object $D$ again (with deleted text).
What is Object Equivalence?

Real-Life Word Processing Example

Click “¶”.

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong. He is a Steering Committee Chair of the IEEE International Conference on Software Quality, Reliability & Security.
What is Object Equivalence?

Real-Life Word Processing Example

T.H. Tse is an Honorary Professor in Computer Science at The University of Hong Kong, a Steering Committee Chair of the International Conference on Software Quality, Reliability & Security.
**What is Object Equivalence?**

**Observational Equivalence**

- Two objects will be *observationally equivalent* if they have the exactly the same *visible attributes* and *behavior*.
What is Object Equivalence?

Observational Equivalence

- Ideal criterion in object-oriented software testing
- **But** very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very very difficult to verify:
  - Need to check an infinite combination of possible behavior for every test case.
Cannot distinguish between visible and hidden attributes and behavior

Cannot distinguish between attributive and observational equivalence
Algebraic Specification

- A ground term is a sequence of operations
  - show(HKU QRS hide(TSE))

- An axiom is a rule that specifies the refinement of a term
  - hide(D D') = hide(D) hide(D')
  - show(D D') = show(D) show(D')
  - show(hide(D)) = show(D)
  - show(W) = W .
Algebraic Specification

- A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a normal form
A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a *normal form*

\[
\text{show(HKU QRS hide(TSE))}
\]
Algebraic Specification

- A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a normal form

\[
\text{show}((\text{HKU} \ \text{QRS} \ \text{hide}(\text{TSE}))) \rightarrow \text{show}((\text{HKU})) \ \text{show}((\text{QRS})) \ \text{show}((\text{hide}(\text{TSE})))
\]

\[
\text{show}(D \ D') = \text{show}(D) \ \text{show} \ (D')
\]
Algebraic Specification

- A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a *normal form*

\[
\text{show}(\text{HKU QRS hide(TSE)})
\]

\[\Rightarrow \text{show}(\text{HKU}) \text{ show}(\text{QRS}) \text{ show}(\text{hide(TSE)})\]

\[\Rightarrow \text{show}(\text{HKU}) \text{ show}(\text{QRS}) \text{ show}(\text{TSE})\]

\[
\text{show}(\text{hide(D)}) = \text{show}(\text{D})
\]
A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a normal form

\[
\text{show}(\text{HKU QRS hide(TSE)}) \\
\Rightarrow \text{show}(\text{HKU}) \text{ show}(\text{QRS}) \text{ show}(\text{hide(TSE)}) \\
\Rightarrow \text{show}(\text{HKU}) \text{ show}(\text{QRS}) \text{ show}(\text{TSE}) \\
\Rightarrow \text{HKU QRS TSE}
\]

\[
\text{show}(W) = W
\]
A ground term may be transformed into another using axioms as left-to-right rewrite rules, until it reaches a normal form

\[ \text{show(HKU QRS hide(TSE))} \Rightarrow \text{show(HKU) show(QRS) show(hide(TSE))} \Rightarrow \text{show(HKU) show(QRS) show(TSE)} \Rightarrow \text{HKU QRS TSE} \]

Normal form.
Algebraic Specification

Denotational Semantics

◆ Mathematical meaning
◆ Like simultaneous equations

\[2x + 3y = 4\]
\[5x + 6y = 7\]

- “=” means “equals”
- The order of the equations is not important.
Algebraic Specification
Operational Semantics

- How the statements should be executed
- Like C programs
  
  ```
  j = 1;
  j = j + 2;
  ```
  - Replace the variable on the left-hand side by the value on the right-hand side.
An algebraic specification is *canonical* if every sequence of rewrites of a ground term produces a unique normal form.
An algebraic specification is *canonical* if every sequence of rewrites of a ground term produces a unique normal form:

\[
\text{show}(\text{HKU}) \; \text{show}(\text{QRS}) \; \text{show}(\text{hide}(\text{TSE}))
\]

\[\rightarrow \text{show}(\text{HKU}) \; \text{show}(\text{QRS}) \; \text{show}(\text{TSE})\]

\[\rightarrow \text{HKU} \; \text{QRS} \; \text{TSE}\]
A canonical algebraic specification is defined as follows: an algebraic specification is *canonical* if every sequence of rewrites of a ground term produces a unique normal form.

Let $show(HKU)$, $show(QRS)$, and $show(hide(TSE))$ be terms in a specification, then:

- $show(HKU) \; show(QRS) \; show(hide(TSE)) \rightarrow show(HKU) \; show(QRS) \; show(TSE)$
- $HKU \; QRS \; TSE$
- $show(HKU) \; show(QRS) \; show(hide(TSE)) \rightarrow HKU \; QRS \; show(hide(TSE))$
- $HKU \; QRS \; TSE$
- $show(HKU) \; show(QRS) \; show(hide(TSE)) \rightarrow HKU \; QRS \; TSE$
For a *canonical* specification, the operational semantics agrees with the denotational semantics.
An implementation $P$ is correct with respect to the specification $Sp$ if and only if
An implementation $P$ is correct with respect to the specification $Sp$ if and only if

- $P$ satisfies the set $(AE)$ of all *attributively equivalent* pairs of ground terms in $Sp$

For example, $\text{hide(TSE)}$ and $\text{delete(TSE)}$ are attributively equivalent.
Target of OO Software Testing

According to Huo Yan Chen et al. (1998)
An implementation $P$ is correct with respect to the specification $Sp$ if and only if:

- **Equivalence Criterion**
  $P$ satisfies the set $(OE)$ of all observationally equivalent pairs of ground terms in $Sp$
Target of OO Software Testing
According to Huo Yan Chen et al. (1998)

- An implementation $P$ is correct with respect to the specification $Sp$ if and only if:
  - *Equivalence Criterion*
    $P$ satisfies the set $(OE)$ of all observationally equivalent pairs of ground terms in $Sp$
  - *Non-Equivalence Criterion*
    $P$ satisfies the set $(OE')$ of all observationally nonequivalent pairs of ground terms in $Sp$.

For example, $\text{hide(TSE)}$ and $\text{delete(TSE)}$ are observationally nonequivalent.
Targets of OO Software Testing

\[ AE \]

Real-world short-cut, but too weak to be useful

\[ ? \]

A jungle of proposals by various researchers.

\[ OE \cup OE' \]

Real-world correctness, but very very very very very very very very difficult to test.
Target of OO Software Testing


- An implementation $P$ is correct with respect to the specification $Sp$ if and only if
  - $P$ satisfies the set ($GI$) of all ground instances of every axiom in $Sp$
An implementation $P$ is correct with respect to the specification $Sp$ if and only if:

- **Equivalence Criterion**

  $P$ satisfies the set ($RP$) of all “equivalent” ground pairs such that one can be rewritten to the other using axioms in $Sp$ as left-to-write rewrite rules.
An implementation $P$ is correct with respect to the specification $Sp$ if and only if:

- **Equivalence Criterion**
  $P$ satisfies the set ($RP$) of all “equivalent” ground pairs such that one can be rewritten to the other using axioms in $Sp$ as left-to-write rewrite rules.

- **Non-Equivalence Criterion**
  $P$ satisfies the set ($RP'$) of all “nonequivalent” ground pairs.
An implementation $P$ is correct with respect to the specification $S_p$ if and only if

$P$ satisfies the set $(FP)$ of all fundamental pairs in $S_p$
An implementation $P$ is correct with respect to the specification $Sp$ if and only if:

- **Equivalence Criterion**
  
  $P$ satisfies the set ($NE$) of all *normally equivalent* ground pairs in $Sp$.
An implementation $P$ is correct with respect to the specification $S_p$ if and only if:

- **Equivalence Criterion**
  $P$ satisfies the set ($NE$) of all *normally equivalent* ground pairs in $S_p$

- **Non-Equivalence Criterion**
  $P$ satisfies the set ($NE'$) of all *normally nonequivalent* ground pairs in $S_p$
An implementation $P$ is correct with respect to the specification $Sp$ if and only if

- $P$ satisfies the set $(CI)$ of all ground instances of every axiom in $Sp$ that contains creators or constructors only.
Targets of OO Software Testing

- Comparisons through empirical studies?
What is Empirical Study?

- Research based on experimentation or observation to answer a specific question or to test a hypothesis
- Undue emphasis in software engineering, even in first-tier publication venues.
What is Empirical Study?

- Statistical significance does not mean research significance

- A hypothesis may be very probable simply because it tells us nothing, or very little

Sir Karl Popper, Professor of Logic and Scientific Method, London School of Economics (1949–1969)
What is Empirical Study?

- Statistical significance does not mean research significance

  - A hypothesis may be very probable simply because it tells us nothing, or very little
  - A high degree of probability is therefore not an indication of “goodness”

Sir Karl Popper, Professor of Logic and Scientific Method, London School of Economics (1949–1969)
What is Empirical Study?

- Statistical significance does not mean research significance
- A hypothesis may be very probable simply because it tells us nothing, or very little
- A high degree of probability is therefore not an indication of “goodness”
- It may be merely a symptom of low informative content.

Sir Karl Popper, Professor of Logic and Scientific Method, London School of Economics (1949–1969)
Presentation Outline

- Background
- Many facets of the test oracle problem
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result
  - Expected outcome = actual execution result

- Jungle of proposals
- Empirical studies?
- What do other researchers do?
- Trim the tree or tame the forest?
What Do Other Researchers Do?

General Relativity

- luminiferous aether
- corpuscular model
- Lorentz’s theory of electrons
- electromagnetic mass
- absolute space and time
- light constancy
- principle of relative motion
- gravitational time dilation
- hole argument
- general covariance
- gravitational redshift
- field equations

◆ A jungle of proposals
So many people today — and even professional scientists — seem to me like somebody who has seen thousands of trees but has never seen a forest.

Albert Einstein
The Collected Papers of Albert Einstein
Princeton University Press
What Do Other Researchers Do?

General Relativity

◆ Trim the individual trees?
◆ Tame the forest!

What Do Others Do?
General Relativity

- luminiferous aether
- corpuscular model
- Lorentz’s theory of electrons
- electromagnetic mass
- absolute space and time
- light constancy
- principle of relative motion
- gravitational time dilation
- hole argument
- general covariance
- gravitational redshift
- field equations

◆ A jungle of proposals

space time curvature

Albert Einstein
## What Do Other Researchers Do?

### Spectrum-Based Fault Localization

- **Risk evaluation formulas**

<table>
<thead>
<tr>
<th>Method</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMPLE2</td>
<td></td>
</tr>
<tr>
<td>Anderberg</td>
<td></td>
</tr>
<tr>
<td>Arithmetic Mean</td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td></td>
</tr>
<tr>
<td>CBI Inc.</td>
<td></td>
</tr>
<tr>
<td>Cohen</td>
<td></td>
</tr>
<tr>
<td>Dice</td>
<td></td>
</tr>
<tr>
<td>Euclid</td>
<td></td>
</tr>
<tr>
<td>Fleiss</td>
<td></td>
</tr>
<tr>
<td>Goodman</td>
<td></td>
</tr>
<tr>
<td>Hamann</td>
<td></td>
</tr>
<tr>
<td>Hamming, etc.</td>
<td></td>
</tr>
<tr>
<td>Jaccard</td>
<td></td>
</tr>
<tr>
<td>Kulczynski2</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td></td>
</tr>
<tr>
<td>Naish1</td>
<td></td>
</tr>
<tr>
<td>Naish2</td>
<td></td>
</tr>
<tr>
<td>Ochiai</td>
<td></td>
</tr>
<tr>
<td>Rogers &amp; Tanimoto</td>
<td></td>
</tr>
<tr>
<td>Rogot1</td>
<td></td>
</tr>
<tr>
<td>Russel &amp; Rao</td>
<td></td>
</tr>
<tr>
<td>Scott</td>
<td></td>
</tr>
<tr>
<td>Simple Matching</td>
<td></td>
</tr>
<tr>
<td>Sokal</td>
<td></td>
</tr>
<tr>
<td>Sørensen-Dice</td>
<td></td>
</tr>
<tr>
<td>Tarantula</td>
<td></td>
</tr>
<tr>
<td>Wong1</td>
<td></td>
</tr>
<tr>
<td>Wong2</td>
<td></td>
</tr>
<tr>
<td>Wong3</td>
<td></td>
</tr>
</tbody>
</table>
What Do Other Researchers Do?

Spectrum-Based Fault Localization

- Another jungle of proposals
- Comparisons through empirical studies in *IEEE TSE, ICSE, and FSE.*
What Do Other Researchers Do?
Spectrum-Based Fault Localization

- Trim the individual trees?
- Tame the forest!

What Do Others Do?
Spectrum-Based Fault Localization

- Risk evaluation formulas
  - AMPLEx
  - Anderberg
  - Arithmetic Mean
  - Binary
  - CBI Inc.
  - Cohen
  - Dice
  - Euclid
  - Fleiss
  - Goodman
  - Hamann
  - Hamming, etc.
  - Jaccard
  - Kulczynski2
  - M2
  - Naish1
  - Naish2
  - Ochiai
  - q_r
  - Rogers & Tanimoto
  - Rogot1
  - Russel & Rao
  - Scott
  - Simple Matching
  - Sokal
  - Sorensen-Dice
  - Tarantula
  - Wong1
  - Wong2
  - Wong3

T.Y. Chen and Team
Spectrum-Based Fault Localization

According to T.Y. Chen and Team (2013)

- Theoretical framework to compare risk evaluation formulas for single-fault programs
- No single formula can outperform the rest
- Among the formulas under study, only five are "maximal"
- Most of best-known formulas are not among them.
Test Case Generation Techniques
According to T.Y. Chen and Team (2008)

- Prove that no test case generation technique can be better than random testing by more than 50%
  - in the absence of further information on possible locations of failure-causing inputs
- Adaptive random testing is close to this theoretic limit.
Partition Testing Techniques

According to T.Y. Chen and Team (2000)

- Prove that proposed proportional sampling strategy is the only partition testing technique that ensures probability of finding at least one failure is no lower than random testing for any program.
Targets of OO Software Testing

- Trim the individual trees?
- Tame the forest!

Targets of OO Software Testing

- AE
- GI
- RP ∪ RP'
- FP
- OE ∪ OE'
- NE ∪ NE'
- CI

- Comparisons through empirical studies?
Toward Ultimate Target of OO Software Testing
Tame the Forest

- $FP \subset CI \subset GI \subset RP \subset NE \subset OE \subset AE$

Subset but not equal
Toward Ultimate Target of OO Software Testing

Tame the Forest

- $FP \subset CI \subset GI \subset RP \subset NE \subset OE \subset AE$
- $OE \subset OE \cup OE'$
  Subset but not equal

- Hence, $(P \text{ satisfies } AE) \Rightarrow (P \text{ satisfies } OE) \Rightarrow (P \text{ satisfies } NE) \Rightarrow (P \text{ satisfies } RP) \Rightarrow (P \text{ satisfies } GI) \Rightarrow (P \text{ satisfies } CI) \Rightarrow (P \text{ satisfies } FP)$ and

- $(P \text{ satisfies } OE \cup OE') \Rightarrow (P \text{ satisfies } OE)$

Not helpful.
Ultimate Target of OO Software Testing

Tame the Forest

- Have we made full use of algebraic specifications?
- Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$
  - $(P$ satisfies $AE)$
Have we made full use of algebraic specifications?

Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$

- $(P$ satisfies $AE)$
  $\iff (P$ satisfies $OE)$

---

**Ultimate Target of OO Software Testing**

**Tame the Forest**
Ultimate Target of OO Software Testing
Tame the Forest

- Have we made full use of algebraic specifications?
- Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$
  - $(P \text{ satisfies } AE) \iff (P \text{ satisfies } OE) \iff (P \text{ satisfies } OE')$
Tame the Forest

- Have we made full use of algebraic specifications?
- Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$
  - $(P \text{ satisfies } AE) \iff (P \text{ satisfies } OE) \iff (P \text{ satisfies } OE') \iff (P \text{ satisfies } OE \cup OE')$
Ultimate Target of OO Software Testing

Tame the Forest

- Have we made full use of algebraic specifications?
- Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$

  - $(P$ satisfies $AE)$
    - $\iff (P$ satisfies $OE) \iff (P$ satisfies $OE')$
    - $\iff (P$ satisfies $OE \cup OE')$
    - $\iff (P$ satisfies $NE) \iff (P$ satisfies $RP)$
    - $\iff (P$ satisfies $GI) \iff (P$ satisfies $CI)$
    - $\iff (P$ satisfies $FP)$. 
**Ultimate Target of OO Software Testing**

**Tame the Forest**

**In short**

- Given a *canonical* specification $Sp$ with *proper imports* and a *complete* implementation $P$
  
  $$(P \text{ satisfies } AE) \iff (P \text{ satisfies } OE \cup OE')$$

- Real-world short-cut considered too weak to be useful
- Real-world correctness considered too difficult to test.
Many Facets of the Test Oracle Problem

Challenge 4:

- Expected outcome = actual execution result
Ubiquitous Computing

- Computing everywhere and at any time
- Applications operate in dynamic environment
Ubiquitous Computing

- Computing everywhere and at any time
- Applications operate in dynamic environment
- Expected outcome = actual execution result
Ubiquitous Computing

- Computing everywhere and at any time
- Applications operate in dynamic environment
- Expected outcome = actual execution result
- When does a test case end?
Ubiquitous Computing

- Computing everywhere and at any time
- Applications operate in dynamic environment
- Expected outcome = actual execution result
- When does a test case end?
  - Middleware remains active and environment context keeps changing!
Ubiquitous Computing
Expected and Actual Context Trends

- Expected Result
- Actual result

Any difference?
Ubiquitous Computing

- Computing everywhere and at any time
- Applications operate in dynamic environment
- Expected outcome = actual execution result
- When does a test case end?
  - Middleware remains active and environment context keeps changing!
- Identify checkpoints where the system is momentarily stable.
Conclusion

- Many facets of the test oracle problem
Conclusion

- Many facets of the test oracle problem
  - **Expected outcome** = actual execution result

Metamorphic testing

Many facets of metamorphic testing.
Many facets of the test oracle problem

- Expected outcome = actual execution result
- Expected outcome = actual execution result

Algebraic specifications

Many facets of algebraic specifications.
Conclusion

Many facets of the test oracle problem

- Expected outcome = actual execution result
- Expected outcome = actual execution result
- Expected outcome = actual execution result

Many facets of ubiquitous computing.
Conclusion

- Jungle of problems
- Jungle of proposals
- Empirical studies are just an exploratory first step rather than the ultimate goal
- Tame the forest rather than trimming individual trees.
Your Comments are Welcome
Thank you