1 Introduction
   • Motivation
   • Preliminaries
   • Limitations of potential coverage

2 Evaluating actual coverage
   • Conditional branching probabilities
   • Coverage probabilities
   • Fault coverage
   • Actual coverage

3 Predicting actual coverage
   • Actual coverage of test cases
   • Expected actual coverage

4 Test suites

5 Conclusions and future work
Motivation for research on testing

- Software is getting more and more complex
- Bugs cost a lot of money
- Testing is an important validation technique in software development
Motivation for research on testing

- Software is getting more and more complex
- Bugs cost a lot of money
- Testing is an important validation technique in software development

Motivation for research on test coverage

- Testing is inherently incomplete
- A notion of quality of a test suite is necessary
- Quantitative evaluation: how good is a test suite?
- ‘Amount’ of specification / implementation examined by a test suite
Intuition about coverage
Intuition about coverage
Existing approaches to coverage

Early work on coverage: code coverage

- Statement coverage
- Condition coverage
- Path coverage
Existing approaches to coverage

Early work on coverage: code coverage

- Statement coverage
- Condition coverage
- Path coverage

Disadvantages: - all faults are considered of equal severity
Existing approaches to coverage

Early work on coverage: code coverage

- Statement coverage
- Condition coverage
- Path coverage

Disadvantages:
- all faults are considered of equal severity
- syntactic point of view

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

System considered as black box

Semantic point of view

Evaluating and Predicting Actual Test Coverage

Introduction

June 5, 2008 5 / 37
Existing approaches to coverage

Early work on coverage: code coverage

- Statement coverage
- Condition coverage
- Path coverage

Disadvantages:
- all faults are considered of equal severity
- syntactic point of view
- different implementation, different coverage
Recipe 1: vegetable soup

- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while
Recipe 1: vegetable soup

- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while
Recipe 1: vegetable soup

- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \( \frac{5}{6} \cdot 10 = 8.33 \)
Recipe 1: vegetable soup
- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Recipe 2: vegetable soup
- Chop an union
- Slice a few carrots
- Slice a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \( \frac{5}{6} \cdot 10 = 8.33 \)
Recipe 1: vegetable soup
- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \( \frac{5}{6} \cdot 10 = 8.33 \)

Recipe 2: vegetable soup
- Chop an union
- Slice a few carrots
- Slice a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while
Recipe 1: vegetable soup
- Chop an union
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \(\frac{5}{6} \cdot 10 = 8.33\)

Recipe 2: vegetable soup
- Chop an union
- Slice a few carrots
- Slice a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \(\frac{5}{7} \cdot 10 = 7.14\)
Recipe 1: vegetable soup
- Chop an onion
- Slice a few carrots and a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \( \frac{5}{6} \cdot 10 = 8.33 \)

Recipe 2: vegetable soup
- Chop an onion
- Slice a few carrots
- Slice a mushroom
- Boil one liter of water
- Add some Maggi
- Put everything in the water
- Wait a while

Quality: \( \frac{5}{7} \cdot 10 = 7.14 \)

Semantic point of view: how does it taste
Existing approaches to coverage

Early work on coverage: code coverage

- Statement coverage
- Condition coverage
- Path coverage

Disadvantages: - all faults are considered of equal severity
- syntactic point of view
- different implementation, different coverage

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Error weights
An LTS is a tuple $A = \langle S, s^0, L, \Delta \rangle$, with:

- $S$ a set of states
- $s^0$ the initial state
- $L$ a set of actions (partitioned into input and output actions)
- $\Delta$ the transition relation (assumed deterministic)
Definition LTSs

An LTS is a tuple $A = \langle S, s^0, L, \Delta \rangle$, with

- $S$ a set of states
- $s^0$ the initial state
- $L$ a set of actions (partitioned into *input* and *output* actions)
- $\Delta$ the transition relation (assumed deterministic)
Preliminaries – Test cases for LTSs

Specification:

\[
\begin{align*}
\delta & \quad 20\text{ct?} \quad 10\text{ct?} \\
\text{s}_1 & \quad s_0 \quad s_2 \\
\text{coffee!} & \quad \text{tea!}
\end{align*}
\]
Preliminaries – Test cases for LTSs

Specification:

Test case:
Preliminaries – Test cases for LTSs

Specification:

- Perform an input

Test case:
Preliminaries – Test cases for LTSs

Specification:

- Perform an input
- Observe all outputs

Test case:

- Evaluating and Predicting Actual Test Coverage
Preliminaries – Test cases for LTSs

Specification:

- Perform an input
- Observe all outputs
- Always stop after an error

Test case:

- Evaluating and Predicting Actual Test Coverage

Introduction

June 5, 2008 9 / 37
Preliminaries – Weighted fault models (WFM)

\[
\begin{align*}
\delta(f(\text{coffee}!)) &= 10 \\
\delta(f(\text{10ct? tea}!)) &= 0 \\
\delta(f(\text{10ct? coffee}!)) &= 5 \\
\delta(f(\text{10ct? } \delta)) &= 4 \\
\delta(f(\text{10ct? tea}! \text{ 10ct? coffee}!)) &= 3 \\
\delta(f(\text{10ct? tea}! \text{ 20ct? } \delta)) &= 3 \\
\delta(f(\text{10ct? tea}! \text{ 20ct? tea}!)) &= 2
\end{align*}
\]

Restriction on weighted fault models

\[0 < \sum_{\sigma \in L^*} f(\sigma) < \infty\]
Preliminaries – Weighted fault models (WFM)

\[ \delta f(\text{coffee!}) = 10 \]

\[ f(\text{coffee!}) = 0 \]

\[ f(\text{tea!}) = 5 \]

\[ f(\delta) = 4 \]

\[ f(\text{tea!} \text{ coffee!}) = 3 \]

\[ f(\text{tea!} \text{ coffee!}) = 3 \]

\[ f(\text{tea!} \text{ tea!}) = 2 \]

Restriction on weighted fault models

\[ 0 < \sum_{\sigma \in \mathcal{L}} f(\sigma) < \infty \]
Preliminaries – Weighted fault models (WFMIs)

Evaluating and Predicting Actual Test Coverage

Introduction

June 5, 2008 10 / 37
Preliminaries – Weighted fault models (WFMs)

\[ f(\text{coffee}!) = 10 \]
\[ f(10\text{ct? tea}!) = 0 \]
\[ f(10\text{ct? coffee}!) = 5 \]
\[ f(10\text{ct?} \delta) = 4 \]
Preliminaries – Weighted fault models (WFMs)

\[ f(\text{coffee!}) = 10 \]
\[ f(10\text{ct? tea!}) = 0 \]
\[ f(10\text{ct? coffee!}) = 5 \]
\[ f(10\text{ct? } \delta) = 4 \]
\[ f(10\text{ct? tea! } 10\text{ct? coffee!}) = 3 \]
Preliminaries – Weighted fault models (WFMs)

\[ f(\text{coffee!}) = 10 \]
\[ f(10\text{ct? tea!}) = 0 \]
\[ f(10\text{ct? coffee!}) = 5 \]
\[ f(10\text{ct? }\delta) = 4 \]
\[ f(10\text{ct? tea! 10ct? coffee!}) = 3 \]
\[ f(10\text{ct? tea! 20ct? }\delta) = 3 \]
\[ f(10\text{ct? tea! 20ct? tea!}) = 2 \]
Restriction on weighted fault models

\[ 0 < \sum_{\sigma \in L^*} f(\sigma) < \infty \]
Preliminaries – Weighted fault models (WFMs)

Evaluating and Predicting Actual Test Coverage

Introduction

June 5, 2008 11 / 37
Preliminaries – Weighted fault models (WFMs)

\[
f(10\text{ct? }\delta) = 4 \\
f(10\text{ct? coffee!}) = 5 \\
f(10\text{ct? tea! }20\text{ct? }\delta) = 3 \\
f(10\text{ct? tea! }20\text{ct? tea!}) = 2
\]
Preliminaries – Weighted fault models (WFMs)

\[
f(10\text{ct? } \delta) = 4 \\
f(10\text{ct? coffee!}) = 5 \\
f(10\text{ct? tea! 20\text{ct? } \delta}) = 3 \\
f(10\text{ct? tea! 20\text{ct? tea!})} = 2
\]
Preliminaries – Weighted fault models (WFMs)

If \( \text{totCov} = 150 \)

\[
\text{absPotCov} = 4 + 5 + 3 + 2 = 14
\]

\[
\text{relPotCov} = \frac{14}{150} = 0.09
\]

Evaluating and Predicting Actual Test Coverage

Introduction

June 5, 2008 12 / 37
Preliminaries – Weighted fault models (WFMs)

If \( \text{totCov} = 150 \)

then

\[
\text{absPotCov} = 4 + 5 + 3 + 2 = 14
\]
If \( totCov = 150 \),
then
\[
absPotCov = 4 + 5 + 3 + 2 = 14
\]
\[
relPotCov = \frac{14}{150} = 0.09
\]
Limitations of potential coverage

Potential coverage

\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]

Actual coverage

Not all faults can be detected at once
Single executions cover only some faults
Executing more often could increase coverage
How many executions are needed?

Necessary to include probabilities!

Evaluating and Predicting Actual Test Coverage
Limitations of potential coverage

Potential coverage

\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]
Limitations of potential coverage

Potential coverage

\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]

Actual coverage

- Not all faults can be detected at once
- Single executions cover only some faults
- Executing more often could increase coverage
- How many executions are needed?
- Necessary to include probabilities!
Limitations of potential coverage

Potential coverage
\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]

Actual coverage
- Not all faults can be detected at once
- Single executions cover only some faults
- Executing more often could increase coverage
- How many executions are needed?
- Necessary to include probabilities!
Limitations of potential coverage

Potential coverage

\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]

Actual coverage

- Not all faults can be detected at once
- Single executions cover only some faults
- Executing more often could increase coverage
- How many executions are needed?
- Necessary to include probabilities!

Potential coverage

\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]
Limitations of potential coverage

Potential coverage
\[ \text{absPotCov}(f, t) = 10 + 15 = 25 \]

Actual coverage
- Not all faults can be detected at once
- Single executions cover only some faults
- Executing more often could increase coverage
- How many executions are needed?
- Necessary to include probabilities!
Contents

1 Introduction
   - Motivation
   - Preliminaries
   - Limitations of potential coverage

2 Evaluating actual coverage
   - Conditional branching probabilities
   - Coverage probabilities
   - Fault coverage
   - Actual coverage

3 Predicting actual coverage
   - Actual coverage of test cases
   - Expected actual coverage

4 Test suites

5 Conclusions and future work
Actual coverage

1. Probabilistic execution model:
   - Branching probabilities ($p^{br}$)
   - Conditional branching probabilities ($p^{cbr}$)
<table>
<thead>
<tr>
<th>Actual coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> Probabilistic execution model:</td>
</tr>
<tr>
<td>- Branching probabilities ($p^{br}$)</td>
</tr>
<tr>
<td>- Conditional branching probabilities ($p^{cbr}$)</td>
</tr>
<tr>
<td><strong>2</strong> Evaluating actual coverage:</td>
</tr>
<tr>
<td>Calculating the actual coverage of a given execution or sequence of executions</td>
</tr>
</tbody>
</table>
1 Probabilistic execution model:
   - Branching probabilities ($p_{br}$)
   - Conditional branching probabilities ($p_{cbr}$)

2 Evaluating actual coverage:
   Calculating the actual coverage of a given execution or sequence of executions

3 Predicting actual coverage:
   Predicting the actual coverage a test case or test suite yields.
Fault coverage

A fault is *covered* by an execution if the execution gives us information about whether the fault is present or absent.
Fault coverage

A fault is covered by an execution if the execution gives us information about whether the fault is present or absent.
Fault coverage

A fault is covered by an execution if the execution gives us information about whether the fault is present or absent.
Fault coverage

A fault is *covered* by an execution if the execution gives us information about whether the fault is present or absent.
Conditional branching probabilities

Evaluating and Predicting Actual Test Coverage

June 5, 2008 17 / 37
Conditional branching probabilities

- **Coffee!**
  - **(1.0) Tea!**
  - **(0.0) Coffee!**
  - **Pass**
  - **Fail**

- **Coffee!**
  - **(0.6) Tea!**
  - **(0.4) Coffee!**
  - **Pass**
  - **Fail**
Conditional branching probabilities

(1.0) tea! coffee! (0.0)

pass fail

coffee! (1.0) tea! coffee! (0.0)

pass fail

coffee! (0.6) tea! coffee! (0.4)

pass fail

(1.0) tea! coffee! (0.0)

pass fail
Conditional branching probabilities

1. (1.0) tea! coffee! (0.0)
   - pass (1.0) 
   - fail (0.0)

2. (1.0) tea! coffee! (0.0)
   - pass (1.0) 
   - fail (0.0)

3. (0.6) tea! coffee! (0.4)
   - pass (0.99) 
   - fail (0.01)

4. (0.99) tea! coffee! (0.01)
   - pass (0.6) 
   - fail (0.4)
Conditional branching probabilities $p^{cbr}$
Coverage probabilities

- Conditional branching probability: 0.4
  \( \mathbb{P}[\text{coffee! produced from blue} | \text{coffee! possible from blue}] \)

\[
\begin{align*}
(1.0) & \quad \text{tea!} \\
(0.0) & \quad \text{coffee!} \\
\text{pass} & \quad \text{fail}
\end{align*}
\]

\[
\begin{align*}
(0.6) & \quad \text{tea!} \\
(0.4) & \quad \text{coffee!} \\
\text{pass} & \quad \text{fail}
\end{align*}
\]
Coverage probabilities

- Conditional branching probability: 0.4
  \[ P[\text{coffee! produced from blue} | \text{coffee! possible from blue}] \]

- Observation: 5 times coffee! tea!

\[
\begin{align*}
(1.0) & \quad \text{tea!} \\
(0.0) & \quad \text{coffee!}
\end{align*}
\]

\[
\begin{align*}
\text{pass} & \quad \downarrow \\
\text{fail} & \quad 
\end{align*}
\]

\[
\begin{align*}
(0.6) & \quad \text{tea!} \\
(0.4) & \quad \text{coffee!}
\end{align*}
\]

\[
\begin{align*}
\text{pass} & \quad \downarrow \\
\text{fail} & \quad 
\end{align*}
\]
Coverage probabilities

- Conditional branching probability: 0.4
  \( \mathbb{P}[\text{coffee! produced from blue} | \text{coffee! possible from blue}] \)

- Observation: 5 times coffee! tea!
- Probability of not even one coffee!:
  \[
  (1 - p^{cbr})^5 = 0.6^5 = 0.08
  \]
Coverage probabilities

- Conditional branching probability: 0.4
  \[ \Pr[\text{coffee! produced from blue} \mid \text{coffee! possible from blue}] \]

- Observation: 5 times coffee! tea!

- Probability of not even one coffee!:
  \[ (1 - p^{cbr})^5 = 0.6^5 = 0.08 \]

- Probability of at least one coffee!:
  \[ 1 - (1 - p^{cbr})^5 = 1 - 0.6^5 = 0.92 \]
Coverage probabilities

- Conditional branching probability: 0.4
  \[ P[\text{coffee! produced from blue} \mid \text{coffee! possible from blue}] \]

- Observation: 5 times coffee! tea!

- Probability of not even one coffee!:
  \[ (1 - p^{cbr})^5 = 0.6^5 = 0.08 \]

- Probability of at least one coffee!:
  \[ 1 - (1 - p^{cbr})^5 = 1 - 0.6^5 = 0.92 \]

- Coverage probability:
  \[ p^{cov} = 1 - (1 - p^{cbr})^k \]
Fault coverage

1. If a fault is shown present, it is *completely covered*.
2. If a fault is shown absent, it is *partially covered*. The coverage probability denotes the fraction.
Fault coverage

1. If a fault is shown present, it is *completely covered*.
2. If a fault is shown absent, it is *partially covered*. The coverage probability denotes the fraction.
Fault coverage

1. If a fault is shown present, it is completely covered.
2. If a fault is shown absent, it is partially covered. The coverage probability denotes the fraction.

Fault coverage

- coffee! coffee! 10
- tea! tea! 0
Fault coverage

1. If a fault is shown present, it is completely covered.
2. If a fault is shown absent, it is partially covered. The coverage probability denotes the fraction.
Fault coverage

1. If a fault is shown present, it is \textit{completely covered}.
2. If a fault is shown absent, it is \textit{partially covered}. The coverage probability denotes the fraction.

\[ p^{\text{cov}}(\text{coffee! coffee!}) = 1 - (1 - 0.4)^3 = 0.78 \]

Fault coverage \textit{coffee! coffee!} 7.8

Fault coverage \textit{tea! tea!} 0
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages

\[
faultCov(b! a? d!) = \]

\[
\text{faultCov}(b! a? c!) = 4.8
\]

\[
\text{faultCov}(d! a? b!) = 0
\]

\[
\text{faultCov}(d! a? d!) = 0
\]

\[
faultCov(c!) = 1.4
\]

\[
absCov = 10.2
\]
Actual coverage

*Actual coverage* of an execution or sequence of executions: The sum of all fault coverages

\[
\text{faultCov}(b! a? d!) = 4
\]
Actual coverage

Actual coverage of an execution or sequence of executions:
The sum of all fault coverages

\[
\begin{align*}
\text{faultCov}(b! a? d!) &= 4 \\
\text{faultCov}(b! a? c!) &= \\
\end{align*}
\]
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages

\[
\begin{align*}
\text{faultCov}(b! a? d!) &= 4 \\
\text{faultCov}(b! a? c!) &= 4.8
\end{align*}
\]

\[
(1 - (1 - 0.8)) \cdot 6 = 4.8
\]
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages

```
faultCov(b! a? d!) = 4
faultCov(b! a? c!) = 4.8
faultCov(d! a? b!) = 
```
Actual coverage

Actual coverage of an execution or sequence of executions:
The sum of all fault coverages

\[
\text{faultCov}(b! a? d!) = 4 \\
\text{faultCov}(b! a? c!) = 4.8 \\
\text{faultCov}(d! a? b!) = 0
\]
Actual coverage of an execution or sequence of executions: The sum of all fault coverages

\[
\text{faultCov}(b! a? d!) = 4
\]
\[
\text{faultCov}(b! a? c!) = 4.8
\]
\[
\text{faultCov}(d! a? b!) = 0
\]
\[
\text{faultCov}(d! a? d!) =
\]
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages

\[ \text{faultCov}(b! a? d!) = 4 \]
\[ \text{faultCov}(b! a? c!) = 4.8 \]
\[ \text{faultCov}(d! a? b!) = 0 \]
\[ \text{faultCov}(d! a? d!) = 0 \]
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages.

\[
\begin{align*}
\text{faultCov}(b! a? d!) &= 4 \\
\text{faultCov}(b! a? c!) &= 4.8 \\
\text{faultCov}(d! a? b!) &= 0 \\
\text{faultCov}(d! a? d!) &= 0 \\
\text{faultCov}(c!) &= 1.4 \\
\end{align*}
\]
Actual coverage

*Actual coverage* of an execution or sequence of executions:
The sum of all fault coverages

\[
\text{faultCov}(b! a? d!) = 4
\]
\[
\text{faultCov}(b! a? c!) = 4.8
\]
\[
\text{faultCov}(d! a? b!) = 0
\]
\[
\text{faultCov}(d! a? d!) = 0
\]
\[
\text{faultCov}(c!) = 1.4
\]

\[
(1 - (1 - 0.2)) \cdot 7 = 1.4
\]
Actual coverage

*Actual coverage* of an execution or sequence of executions: The sum of all fault coverages

\[
\text{absCov} = 10.2
\]

\[
\text{faultCov}(b! a? d!) = 4
\]

\[
\text{faultCov}(b! a? c!) = 4.8
\]

\[
\text{faultCov}(d! a? b!) = 0
\]

\[
\text{faultCov}(d! a? d!) = 0
\]

\[
\text{faultCov}(c!) = 1.4
\]
Contents

1 Introduction
   • Motivation
   • Preliminaries
   • Limitations of potential coverage

2 Evaluating actual coverage
   • Conditional branching probabilities
   • Coverage probabilities
   • Fault coverage
   • Actual coverage

3 Predicting actual coverage
   • Actual coverage of test cases
   • Expected actual coverage

4 Test suites

5 Conclusions and future work
Actual coverage distribution of a test case

The *actual coverage distribution* of a test case predicts its actual coverage.
Actual coverage distribution of a test case

The *actual coverage distribution* of a test case predicts its actual coverage.

---

The diagram shows a decision tree with two branches. The left branch starts with "coffee!" leading to "tea!" and "pass". The right branch starts with "tea!" leading to "coffee!" and "fail". Further down the tree, there are numerical values indicating the probabilities and counts for each outcome.
Actual coverage of test cases

The actual coverage distribution of a test case predicts its actual coverage.

The actual coverage distribution of a test case predicts its actual coverage.

Evaluating and Predicting Actual Test Coverage

June 5, 2008 22 / 37
The actual coverage distribution of a test case predicts its actual coverage.
Actual coverage of test cases

The *actual coverage distribution* of a test case predicts its actual coverage.

### AbsCov

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>(0.4)</td>
</tr>
<tr>
<td>15</td>
<td>(0.4)</td>
</tr>
</tbody>
</table>

### absCov

| 10 |
| 15 |
| 4 |
Actual coverage of test cases

Actual coverage distribution of a test case

The *actual coverage distribution* of a test case predicts its actual coverage.

Evaluating and Predicting Actual Test Coverage
The actual coverage distribution of a test case predicts its actual coverage.

<table>
<thead>
<tr>
<th>absCov</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.015</td>
</tr>
<tr>
<td>15</td>
<td>0.005</td>
</tr>
<tr>
<td>4</td>
<td>0.735</td>
</tr>
<tr>
<td>6</td>
<td>0.245</td>
</tr>
</tbody>
</table>
The *actual coverage distribution* of a test case predicts its actual coverage.

The diagram shows the actual coverage distribution with branches labeled "tea!" and "coffee!" leading to "pass" and "fail" outcomes. The actual coverage values for each outcome are indicated with numbers and probabilities.

<table>
<thead>
<tr>
<th>absCov</th>
<th>P</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.015</td>
<td>0.150</td>
</tr>
<tr>
<td>15</td>
<td>0.005</td>
<td>0.075</td>
</tr>
<tr>
<td>4</td>
<td>0.735</td>
<td>2.940</td>
</tr>
<tr>
<td>6</td>
<td>0.245</td>
<td>1.470</td>
</tr>
</tbody>
</table>
The *actual coverage distribution* of a test case predicts its actual coverage.

\[
\begin{array}{c|c|c|c}
\text{absCov} & \mathbb{P} & \times \\
10 & 0.015 & 0.150 \\
15 & 0.005 & 0.075 \\
4 & 0.735 & 2.940 \\
6 & 0.245 & 1.470 \\
\hline
\mathbb{E}(\text{absCov}) = 4.635 \\
\end{array}
\]
The branching probabilities $p^{br}$ describe how the implementation is expected to behave.
The branching probabilities $p^{br}$ describe how the implementation is expected to behave.
The branching probabilities $p^{br}$ describe how the implementation is expected to behave.

- (0.75) coffee!
- (0.25) tea!

- (0.98) tea! (0.02) coffee!
- (0.02) tea!

- (0.98) coffee! (0.02) tea!
- (0.02) coffee!

- pass (0.4) 10
- fail (0.4) 15
Trace occurrence probabilities and actual coverage

\[
\begin{align*}
&\text{(0.75) coffee!} & \text{tea! (0.25)} \\
&\text{(0.98) tea!} & \text{coffee! (0.02)} & \text{tea!} & \text{coffee! (0.98)} \\
&\text{pass} & \text{fail} & \text{fail} & \text{pass} \\
\end{align*}
\]

\[
\begin{align*}
&\text{pass} \quad \text{fail} \\
&\text{10} \quad \text{15} \\
\end{align*}
\]
Trace occurrence probabilities and actual coverage

\[ p^{to}(\text{coffee! tea!}) = 0.75 \cdot 0.98 = 0.735 \]
Trace occurrence probabilities and actual coverage

\[ p^{to}(coffee! \ tea!) = 0.75 \cdot 0.98 = 0.735 \]
\[ p^{to}(coffee! \ coffee!) = 0.75 \cdot 0.02 = 0.015 \]
\[ p^{to}(tea! \ tea!) = 0.75 \cdot 0.02 = 0.005 \]
\[ p^{to}(tea! \ coffee!) = 0.25 \cdot 0.98 = 0.245 \]
Suppose we perform three executions of

\[(0.75) \text{ coffee!} \quad \text{tea! } (0.25)\]

\[(0.98) \text{ tea!} \quad \text{coffee! } (0.02) (0.02) \text{ tea!} \quad \text{coffee! } (0.98)\]

Possible observation: [blue, blue, red]

Actual coverage:

\[15 + (1 - (1 - 0.4)^2) \cdot 10 = 15 + 6.4 = 21.4\]
Suppose we perform three executions of

- (0.75) coffee!
- (0.25) tea!
- (0.98) tea!
- (0.02) coffee!
- (0.02) tea!
- (0.98) coffee!

Possible observation: [blue, blue, red]
Suppose we perform three executions of:

- (0.75) coffee!
- tea! (0.25)
- (0.98) tea!
- coffee! (0.02)
- x coffee! (0.02)
- tea!
- coffee! (0.98)

Possible observation: [blue, blue, red]

Actual coverage:

\[ 15 + \]
Suppose we perform three executions of

\[
\begin{align*}
(0.75) \text{ coffee!} \\
(0.98) \text{ tea!} \\
\text{pass} \\
\text{fail} \\
10
\end{align*}
\]

\[
\begin{align*}
\text{coffee!} (0.02) (0.02) \text{ tea!} \\
\text{tea!} (0.25) \\
\text{fail} \\
15
\end{align*}
\]

\[
\begin{align*}
\text{coffee!} (0.98) \\
\text{pass} \\
15
\end{align*}
\]

Possible observation: [blue, blue, red]

Actual coverage:

\[
15 + (1 - (1 - 0.4)^2) \cdot 10
\]
Suppose we perform three executions of

(0.75) coffee!
(0.25) tea!

(0.98) tea!
(0.02) coffee!
(0.98) coffee!
(0.02) tea!

Possible observation: [blue, blue, red]
Actual coverage:

\[ 15 + (1 - (1 - 0.4)^2) \cdot 10 = 15 + 6.4 = 21.4 \]
Sequences of executions

Suppose we perform three executions of

Possible observation: [blue, blue, red]

Actual coverage:

$$15 + (1 - (1 - 0.4)^2) \cdot 10 = 15 + 6.4 = 21.4$$

Many observations possible: $O(|\text{exec}|^n)$
Expected actual coverage for a sequence of executions

Theorem

$$E(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \cdot (1 - (1 - p^{cbr}(a | \sigma))^k) \right)$$
Theorem

\[ \mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \cdot (1 - (1 - p^{cbr}(a | \sigma)^k)) \right) \]
Expected actual coverage for a sequence of executions

**Theorem**

\[
\mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( 1 - (1 - p^{to}(\sigma a))^n \right) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k.
\]

\[
\mathbb{E}(p^{cov}(\sigma a)) = \left( 1 - (1 - p^{cbr}(a | \sigma))^k \right)
\]
Theorem

\[ E(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right) \]

Case 1: presence shown

\[ (1 - (1 - p^{cbr}(a | \sigma)^k)) \]
Expected actual coverage for a sequence of executions

Theorem

\[ \mathbb{E}(actCov_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \right. \]
\[ \left. \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right) \]

Case 1: presence shown
\[ \mathbb{P}[\text{observe } \sigma a] \]

(1 - (1 - p^{cbr}(a | \sigma)^k))
Expected actual coverage for a sequence of executions

**Theorem**

\[
\mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right)
\]

**Case 1: presence shown**

\[\mathbb{P}[\text{not observe } \sigma a] \]

(Evaluating and Predicting Actual Test Coverage)
Expected actual coverage for a sequence of executions

**Theorem**

\[
\mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right)
\]

**Case 1: presence shown**

\[\mathbb{P}[\text{never observe } \sigma a] = (1 - (1 - p^{cbr}(a | \sigma)^k))\]
Expected actual coverage for a sequence of executions

**Theorem**

\[ \mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{\text{to}}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{\text{to}}(\sigma)^k (1 - p^{\text{to}}(\sigma))^{n-k} \cdot (1 - p^{\text{br}}(a | \sigma))^k \cdot (1 - (1 - p^{\text{cbr}}(a | \sigma))^k) \right) \]

**Case 1: presence shown**

\[ P[\text{ever observe } \sigma a] \]
Expected actual coverage for a sequence of executions

**Theorem**

\[ E(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right) \]

Case 2: presence not shown
Absence shown 0 \ldots n times
Coverage probabilities

- Conditional branching probability: 0.4
  \[ \mathbb{P}[\text{coffee! produced from blue} \mid \text{coffee! possible from blue}] \]

- Observation: 5 times coffee! tea!
- Probability of not even one coffee!:
  \[ (1 - p^{\text{cbr}})^5 = 0.6^5 = 0.08 \]
- Probability of at least one coffee!:
  \[ 1 - (1 - p^{\text{cbr}})^5 = 1 - 0.6^5 = 0.92 \]
- Coverage probability:
  \[ p^{\text{cov}} = 1 - (1 - p^{\text{cbr}})^k \]
Expected actual coverage for a sequence of executions

**Theorem**

\[
\mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right)
\]

Case 2: presence not shown
Absence shown 0 \ldots n times

\[
(1 - (1 - p^{cbr}(a | \sigma)^k))
\]
Theorem

\[ \mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \right. \\
\left. \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \cdot (1 - (1 - p^{cbr}(a | \sigma)^k)) \right) \]
Expected actual coverage for a sequence of executions

**Theorem**

\[ \mathbb{E}(actCov_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left(1 - (1 - p^{to}(\sigma a))^n\right) \cdot 1 + \]

\[ \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \cdot (1 - (1 - p^{cbr}(a | \sigma)^k)). \]

**Case 2:** presence not shown

\[ \mathbb{P}[k \text{ times } \sigma] \]
Theorem

\[ \mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right) \]

Case 2: presence not shown
\[ P[\text{the others not } \sigma] \]
Theorem

\[ \mathbb{E}(actCov_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a \mid \sigma))^k \right) \]

Case 2: presence not shown
All possible orderings

(1 - (1 - p^{cbr}(a \mid \sigma)^k))
Expected actual coverage for a sequence of executions

**Theorem**

\[ \mathbb{E}(\text{actCov}_n) = \sum_{\sigma a \in \text{err}_t} f(\sigma a) \cdot \left( (1 - (1 - p^{to}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{to}(\sigma)^k (1 - p^{to}(\sigma))^{n-k} \cdot (1 - p^{br}(a | \sigma))^k \right) \]

Case 2: presence not shown
\[ P[\text{no presence shown}] = (1 - (1 - p^{cbr}(a | \sigma)^k)) \]
Example of expected actual coverage

\[
10 \cdot \left( (1 - (1 - 0.75 \cdot 0.02)^5) + \sum_{k=0}^{5} \binom{5}{k} 0.75^k \cdot (1 - 0.75)^{5-k} \cdot (1 - 0.02)^k \cdot (1 - (1 - 0.4)^k) \right) +
15 \cdot \left( (1 - (1 - 0.25 \cdot 0.02)^5) + \sum_{k=0}^{5} \binom{5}{k} 0.25^k \cdot (1 - 0.25)^{5-k} \cdot (1 - 0.02)^k \cdot (1 - (1 - 0.4)^k) \right) = 14.7
\]
Example of expected actual coverage

\[ \mathbb{E}(actCov_1) = 4.6 \]

\[ \mathbb{E}(actCov_5) = 14.7 \]
Example of expected actual coverage

\[ \mathbb{E}(actCov_1) = 4.6 \]
\[ \mathbb{E}(actCov_2) = 8.2 \]
\[ \mathbb{E}(actCov_3) = 10.9 \]
\[ \mathbb{E}(actCov_4) = 13.0 \]
\[ \mathbb{E}(actCov_5) = 14.7 \]
Example of expected actual coverage

\[ \mathbb{E}(actCov_1) = 4.6 \]
\[ \mathbb{E}(actCov_2) = 8.2 \]
\[ \mathbb{E}(actCov_3) = 10.9 \]
\[ \mathbb{E}(actCov_4) = 13.0 \]
\[ \mathbb{E}(actCov_5) = 14.7 \]
\[ \mathbb{E}(actCov_{10}) = 19.7 \]
\[ \mathbb{E}(actCov_{25}) = 24.0 \]
\[ \mathbb{E}(actCov_{50}) = 24.9 \]
Asymptotical behaviour of actual coverage

\[
\lim_{n \to \infty} E(\text{absCov}_n) = \text{absPotCov}
\]

Theorem

Evaluating and Predicting Actual Test Coverage

Predicting actual coverage

June 5, 2008
Asymptotical behaviour of actual coverage

Theorem

\[ \lim_{n \to \infty} \mathbb{E}(\text{absCov}_n) = \text{absPotCov} \]
1 Introduction
   • Motivation
   • Preliminaries
   • Limitations of potential coverage

2 Evaluating actual coverage
   • Conditional branching probabilities
   • Coverage probabilities
   • Fault coverage
   • Actual coverage

3 Predicting actual coverage
   • Actual coverage of test cases
   • Expected actual coverage

4 Test suites

5 Conclusions and future work
Very similar to actual coverage for test cases:
sum all the fault coverages
Take into account in how many test cases an erroneous trace is contained
Again, an efficient formula for the expected actual coverage exists

Theorem
\[
\lim_{n \to \infty} E(\text{absCov}_n) = \text{absPotCov}
\]
Contents

1 Introduction
   • Motivation
   • Preliminaries
   • Limitations of potential coverage

2 Evaluating actual coverage
   • Conditional branching probabilities
   • Coverage probabilities
   • Fault coverage
   • Actual coverage

3 Predicting actual coverage
   • Actual coverage of test cases
   • Expected actual coverage

4 Test suites

5 Conclusions and future work
Main results – what you saw today

- New notion of coverage: *actual coverage*
- Evaluating actual coverage of a given execution
- Predicting actual coverage of a test case or test suite
Conclusions

Main results – what you saw today

- New notion of coverage: *actual coverage*
- Evaluating actual coverage of a given execution
- Predicting actual coverage of a test case or test suite

Main results – what I did not show

- Probabilistic fault automata
- Methods to derive $p^{cbr}$ and $p^{br}$
- Mathematical proofs
- Detailed example
- Extra features:
  - Risk-based testing
  - Alternative approach to coverage probabilities
  - Approximations
Future work

Directions for future work

- Validation of the framework: tool support, case studies
- Dependencies between errors
- Accuracy of approximations
- On-the-fly test derivation