Formal Methods & Tools

University of Twente
Enschede - The Netherlands

Actual Test Coverage for Embedded Systems

Mark Timmer
University of Twente

14th Dutch Testing Day
November 27, 2008

Contents

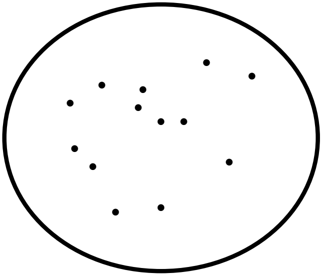
Introduction – coverage

Why coverage?

- Testing is inherently incomplete
- Testing does increase our confidence in the system
- A notion of *quality* of a test suite is necessary
- Coverage: ‘amount’ of specification / implementation examined by a test suite

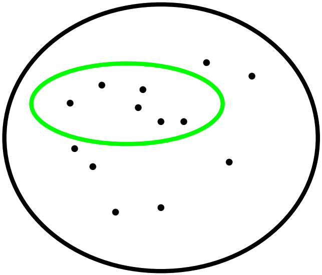
Why coverage?

- Testing is inherently incomplete
- Testing does increase our confidence in the system
- A notion of *quality* of a test suite is necessary
- Coverage: 'amount' of specification / implementation examined by a test suite



Why coverage?

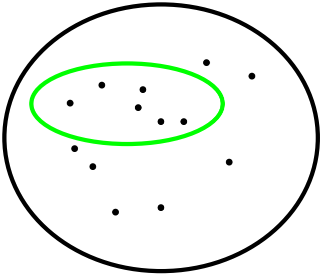
- Testing is inherently incomplete
- Testing does increase our confidence in the system
- A notion of *quality* of a test suite is necessary
- Coverage: 'amount' of specification / implementation examined by a test suite



Introduction – coverage

Why coverage?

- Testing is inherently incomplete
- Testing does increase our confidence in the system
- A notion of *quality* of a test suite is necessary
- Coverage: 'amount' of specification / implementation examined by a test suite



Coverage: $\frac{6}{13} = 46\%$.

Early work on coverage

- Statement coverage
- Path coverage

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations: - all faults are considered of equal severity
- no probabilities
- syntactic point of view

Early work on coverage

- Statement coverage
- Path coverage

Limitations:
- all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations: - all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations: - all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Quality: $\frac{4}{5} \cdot 10 = 8$

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations: - all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Quality: $\frac{4}{5} \cdot 10 = 8$

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations:
- all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Quality: $\frac{4}{5} \cdot 10 = 8$

Introduction – Existing approaches to coverage

Early work on coverage

- Statement coverage
- Path coverage

Limitations:
- all faults are considered of equal severity
- no probabilities
- syntactic point of view

Recipe 1: vegetable soup

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

Quality: $\frac{4}{5} \cdot 10 = 8$

Recipe 2: vegetable soup

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

Quality: $\frac{4}{6} \cdot 10 = 6.7$

Introduction – Semantic 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
- Fault weights

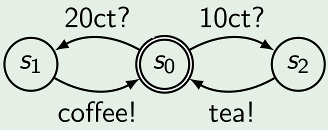
Introduction – Semantic coverage

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems

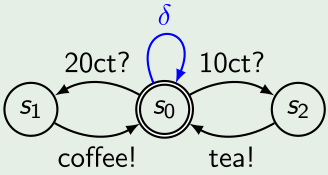


Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems

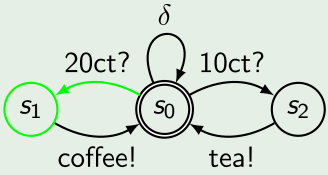


Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



20ct?

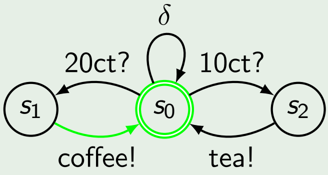
Introduction – Semantic coverage

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



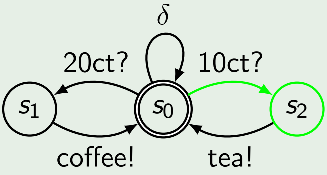
20ct? coffee!

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



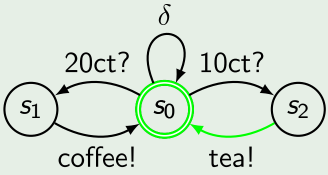
20ct? coffee! 10ct?

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



20ct? coffee! 10ct? tea!

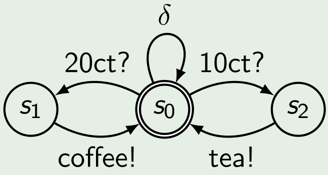
Introduction – Semantic coverage

Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

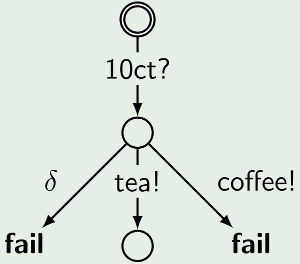
- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



20ct? coffee! 10ct? tea!

Test cases

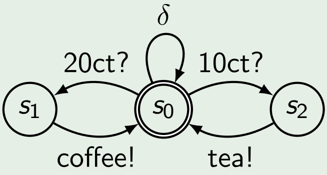


Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

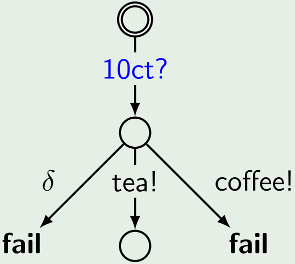
- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



20ct? coffee! 10ct? tea!

Test cases

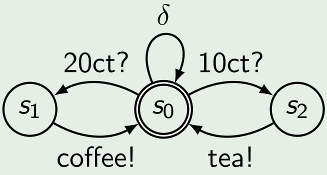


Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

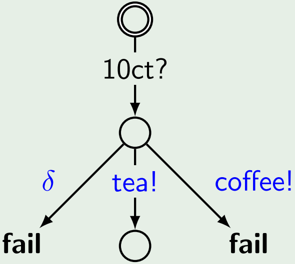
- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems



20ct? coffee! 10ct? tea!

Test cases

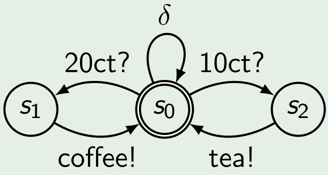


Starting point for my work: semantic coverage

Previous work by Brandán Briones, Brinksma and Stoelinga

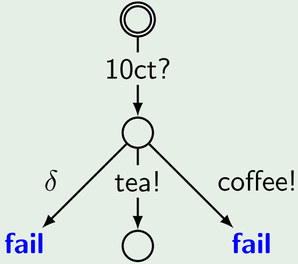
- System considered as black box
- Semantic point of view
- Fault weights

Labelled transition systems

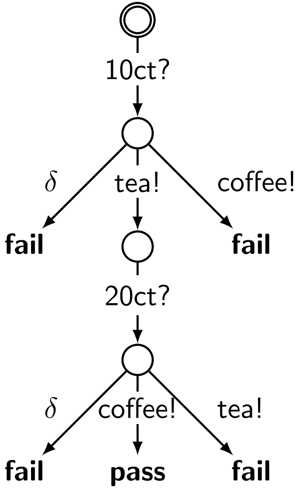


20ct? coffee! 10ct? tea!

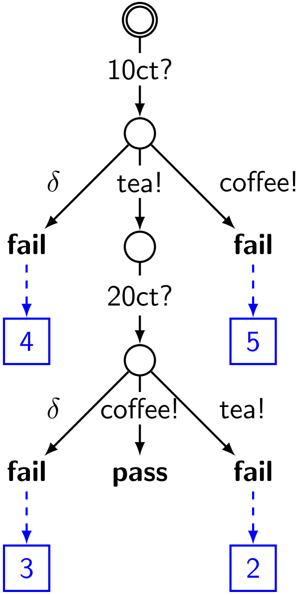
Test cases



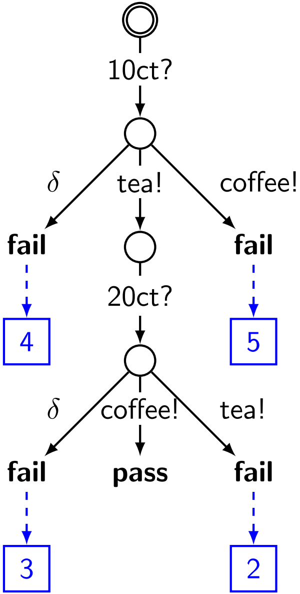
Introduction – Fault weights and coverage measures



Introduction – Fault weights and coverage measures



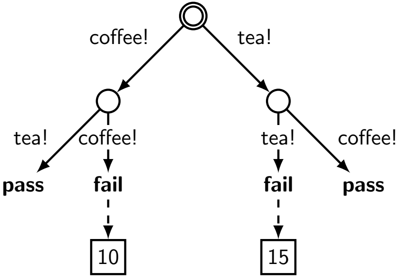
Introduction – Fault weights and coverage measures

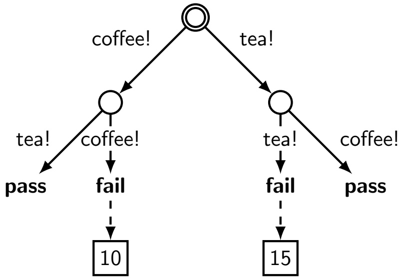


Absolute potential coverage

$4 + 5 + 3 + 2 = 14$

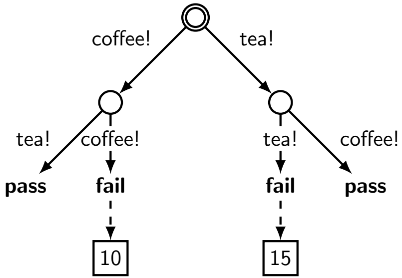
Introduction - Limitations of potential coverage





Absolute potential coverage

$$10 + 15 = 25$$

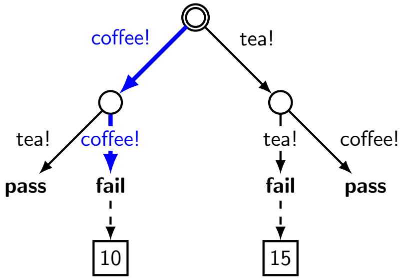


Absolute potential coverage

$$10 + 15 = 25$$

Some remarks

- 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!



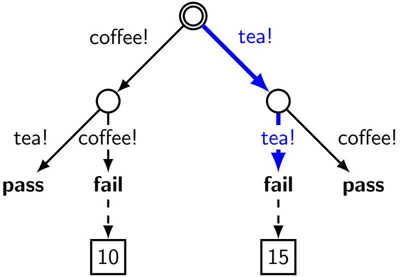
Absolute potential coverage

$$10 + 15 = 25$$

Some remarks

- 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!

Introduction - Limitations of potential coverage



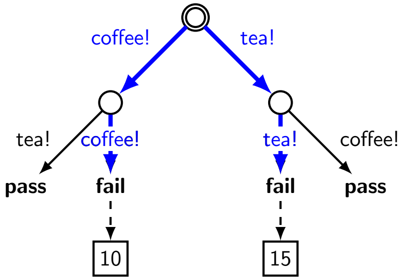
Absolute potential coverage

$$10 + 15 = 25$$

Some remarks

- 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!

Introduction - Limitations of potential coverage



Absolute potential coverage

$$10 + 15 = 25$$

Some remarks

- 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!

Overview of actual coverage

Actual coverage

1. Probabilistic execution model:
   - Branching probabilities (p^{br})
   - Conditional branching probabilities (p^{cbr})

Overview of actual coverage

Actual coverage

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

Overview of actual coverage

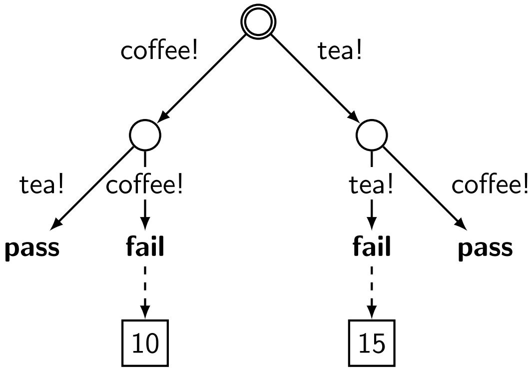
Actual coverage

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.

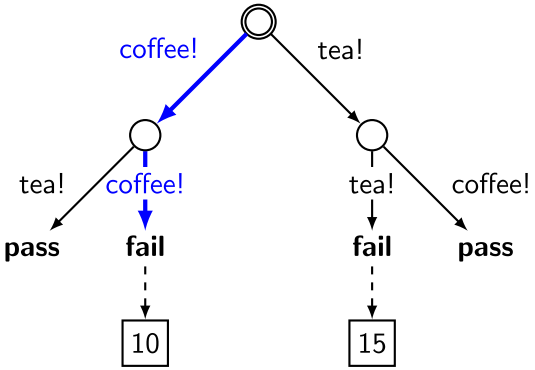
Fault coverage



Fault coverage

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

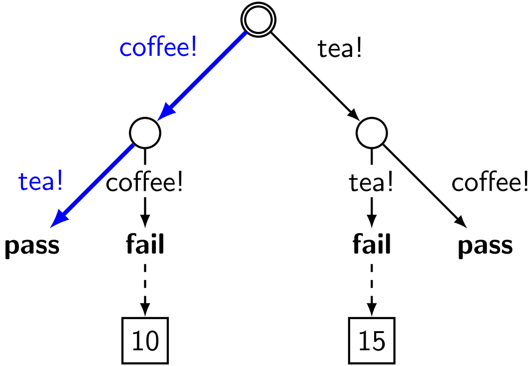
Fault coverage



Fault coverage

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

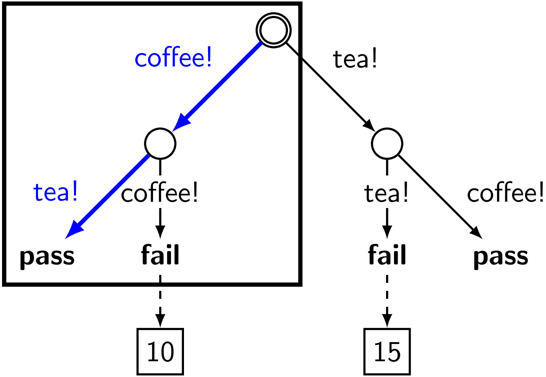
Fault coverage



Fault coverage

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

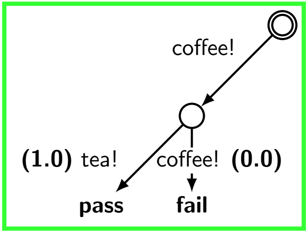
Fault coverage



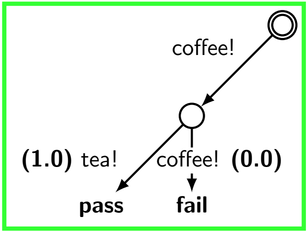
Fault coverage

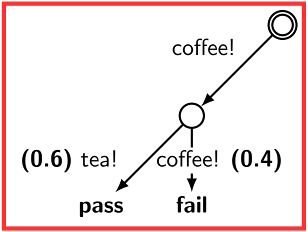
A fault is *covered* by an execution if the execution gives us information about whether the fault is present or absent

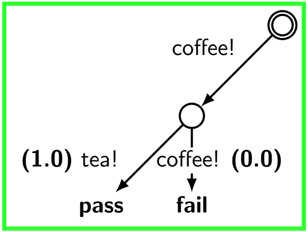
Fault coverage

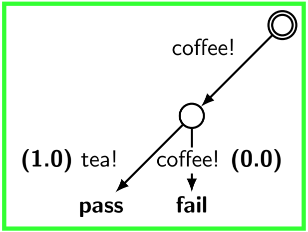


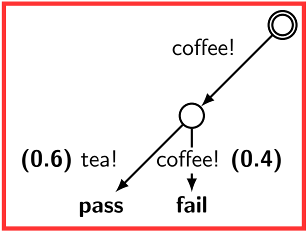
Fault coverage



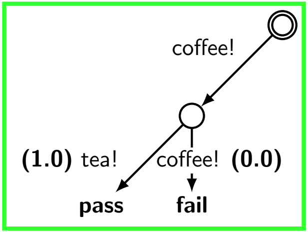


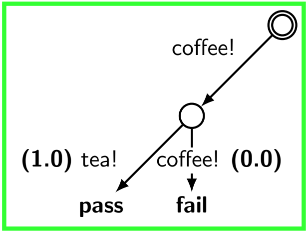

coffee!
(1.0) tea!
coffee! (0.0)
pass
fail

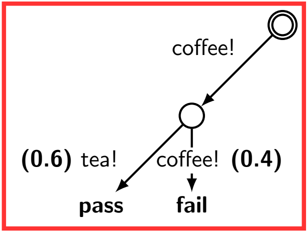

coffee!
(1.0) tea!
coffee! (0.0)
pass
fail

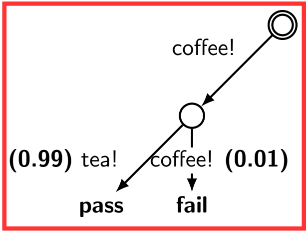

coffee!
(0.6) tea!
coffee! (0.4)
pass
fail

Fault coverage

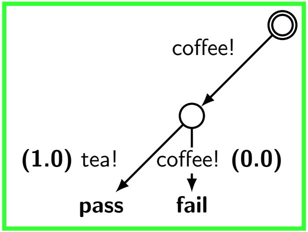


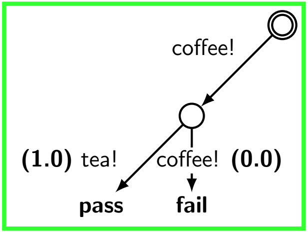


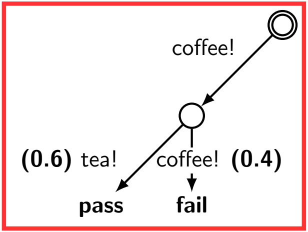


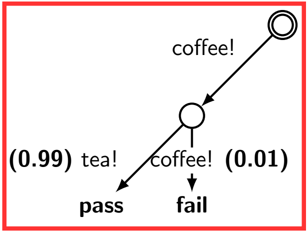


Fault coverage









Conditional branching probabilities p^{cbr}

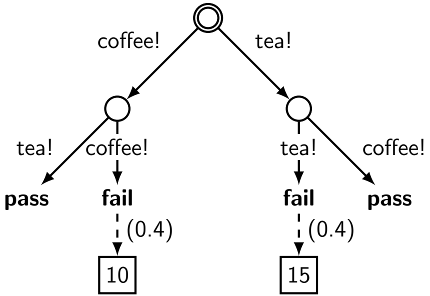
Fault coverage

Fault coverage

1. If a fault is shown present, it is *completely covered*
2. If a fault is shown absent, it is *partially covered*.

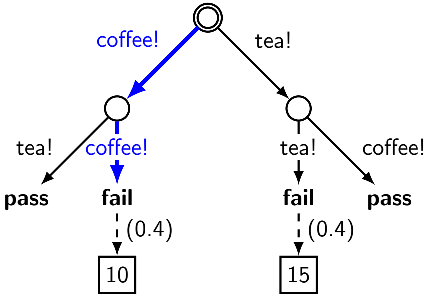
Fault coverage

1. If a fault is shown present, it is *completely covered*
2. If a fault is shown absent, it is *partially covered*.



Fault coverage

1. If a fault is shown present, it is *completely covered*
2. If a fault is shown absent, it is *partially covered*.

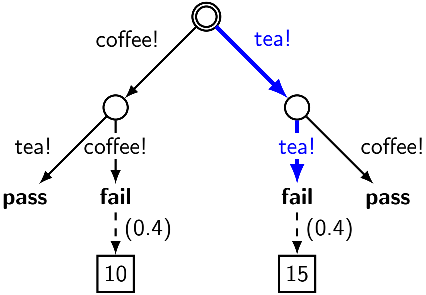


Fault coverage *coffee! coffee!*
10

Fault coverage *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*.

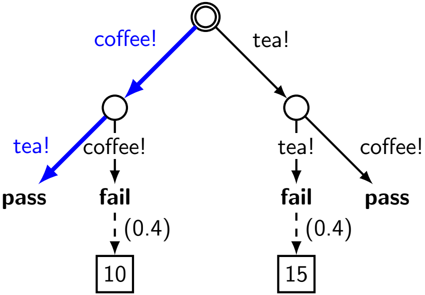


Fault coverage *coffee! coffee!*
0

Fault coverage *tea! tea!*
15

Fault coverage

1. If a fault is shown present, it is *completely covered*
2. If a fault is shown absent, it is *partially covered*.



Fault coverage *coffee! coffee!*
4 – 6.4 – 7.8 – 8.7

Fault coverage *tea! tea!*
0

Evaluating actual coverage

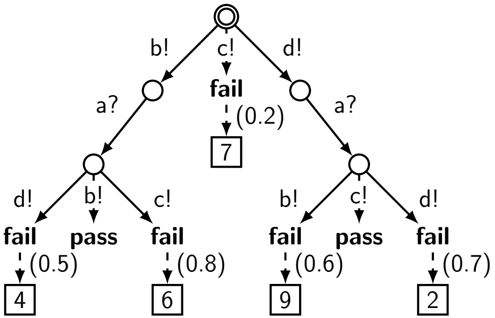
Actual coverage

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

Evaluating actual coverage

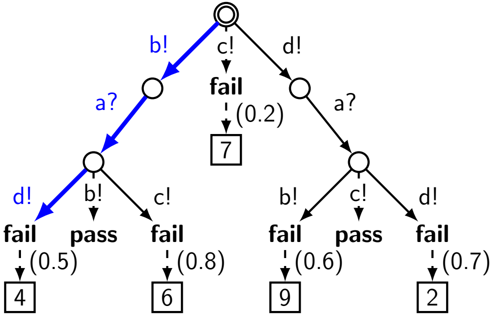
Actual coverage

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



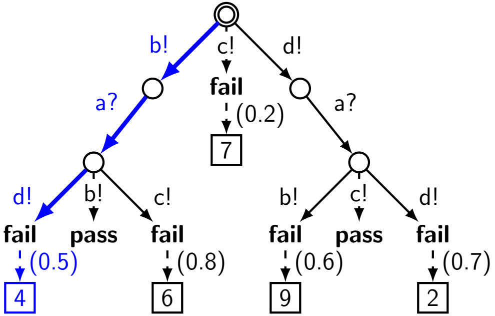
Actual coverage

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



Actual coverage

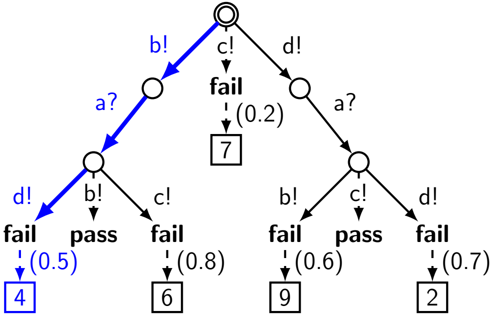
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



faultCov(b! a? d!) =

Actual coverage

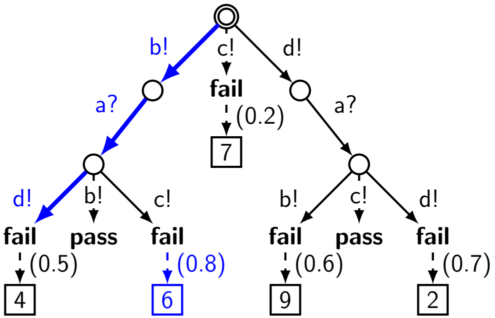
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



$$\mathit{faultCov}(b!\ a?\ d!) = 4$$

Actual coverage

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

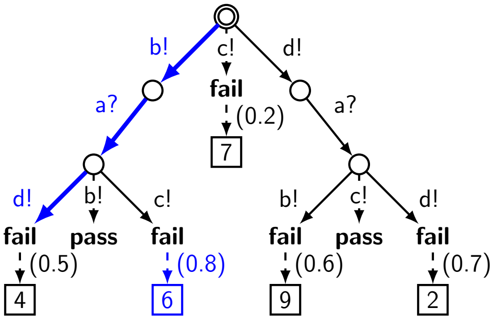


$$
\begin{aligned}
\mathit{faultCov}(b!\ a?\ d!) &= 4 \\
\mathit{faultCov}(b!\ a?\ c!) &=
\end{aligned}
$$

Evaluating actual coverage

Actual coverage

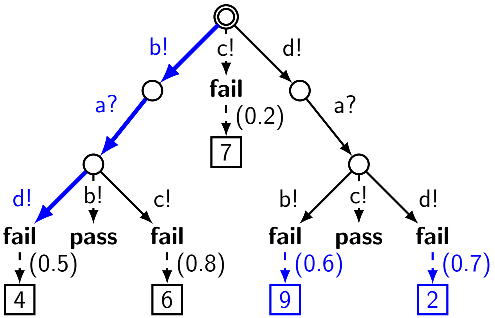
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



$$
\begin{aligned}
\mathit{faultCov}(b!\ a?\ d!) &= 4 \\
\mathit{faultCov}(b!\ a?\ c!) &= 4.8
\end{aligned}
$$

Actual coverage

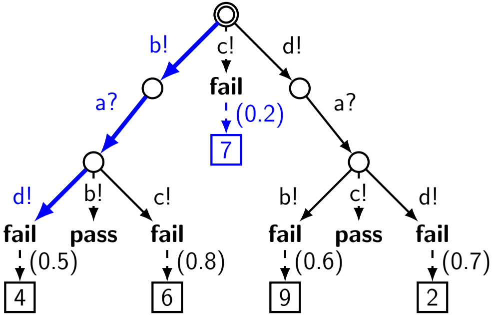
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



$$\begin{aligned} \mathit{faultCov}(b!\ a?\ d!) &= 4 \\ \mathit{faultCov}(b!\ a?\ c!) &= 4.8 \\ \mathit{faultCov}(d!\ a?\ b!) &= 0 \\ \mathit{faultCov}(d!\ a?\ d!) &= 0 \end{aligned}$$

Actual coverage

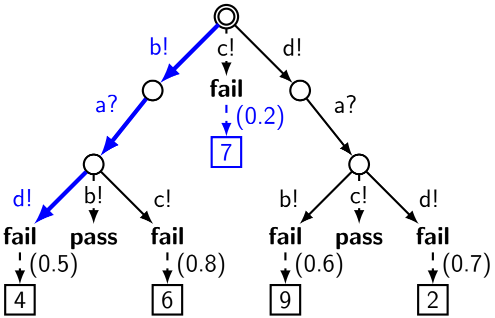
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



$$
\begin{aligned}
faultCov(b!\ a?\ d!) &= 4 \\
faultCov(b!\ a?\ c!) &= 4.8 \\
faultCov(d!\ a?\ b!) &= 0 \\
faultCov(d!\ a?\ d!) &= 0 \\
faultCov(c!) &=
\end{aligned}
$$

Actual coverage

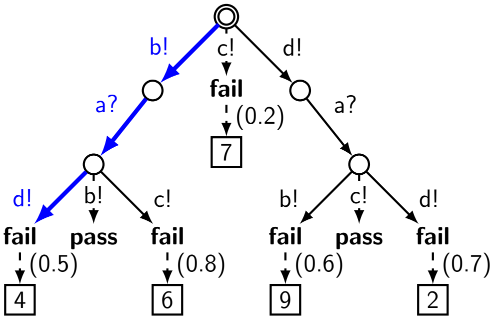
Actual coverage of an execution or sequence of executions:
The sum of all fault coverages



$$
\begin{aligned}
faultCov(b!\ a?\ d!) &= 4\\
faultCov(b!\ a?\ c!) &= 4.8\\
faultCov(d!\ a?\ b!) &= 0\\
faultCov(d!\ a?\ d!) &= 0\\
faultCov(c!) &= 1.4
\end{aligned}
$$

Actual coverage

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



$$
\begin{aligned}
faultCov(b!\ a?\ d!) &= 4\\
faultCov(b!\ a?\ c!) &= 4.8\\
faultCov(d!\ a?\ b!) &= 0\\
faultCov(d!\ a?\ d!) &= 0\\
faultCov(c!) &= 1.4\\
\\
absCov &= 10.2
\end{aligned}
$$

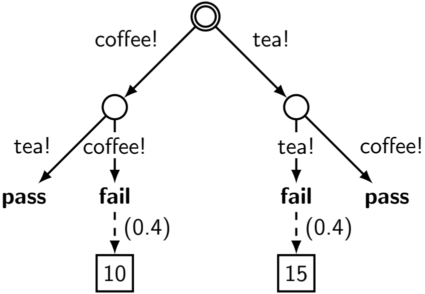
Predicting actual coverage

Actual coverage distribution of a test case

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

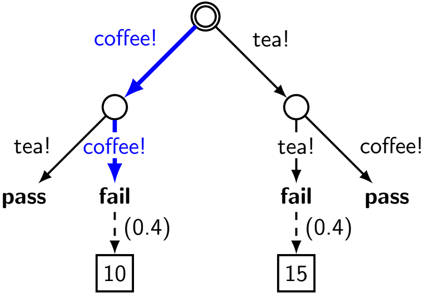
Actual coverage distribution of a test case

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



Actual coverage distribution of a test case

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

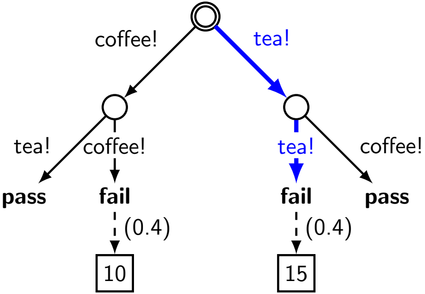


absCov
10

Predicting actual coverage

Actual coverage distribution of a test case

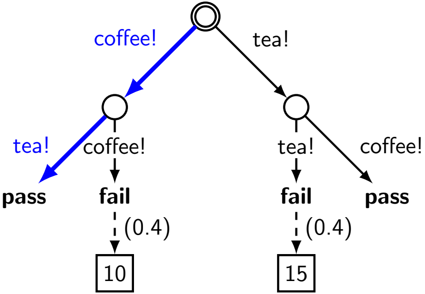
The *actual coverage distribution* of a test case predicts its actual coverage.



absCov
10
15

Actual coverage distribution of a test case

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

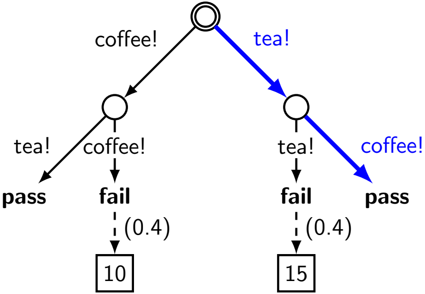


absCov
10
15
4

Predicting actual coverage

Actual coverage distribution of a test case

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

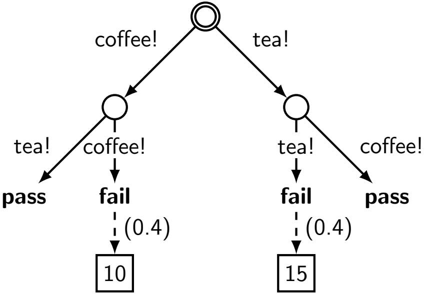


absCov
10
15
4
6

Predicting actual coverage

Actual coverage distribution of a test case

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

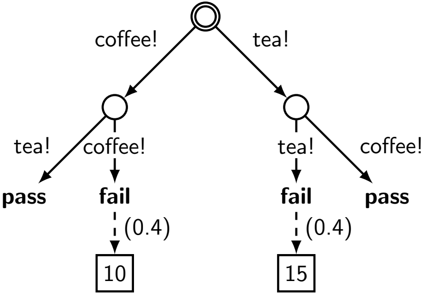


absCov	$\mathbb{P}$
10	0.015
15	0.005
4	0.735
6	0.245

Predicting actual coverage

Actual coverage distribution of a test case

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

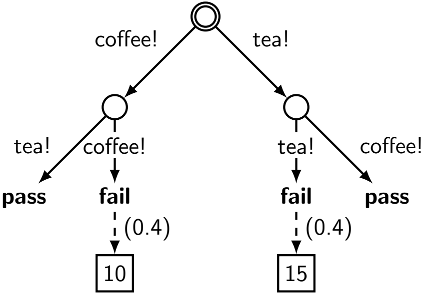


absCov	$\mathbb{P}$	$\times$
10	0.015	0.150
15	0.005	0.075
4	0.735	2.940
6	0.245	1.470

Predicting actual coverage

Actual coverage distribution of a test case

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

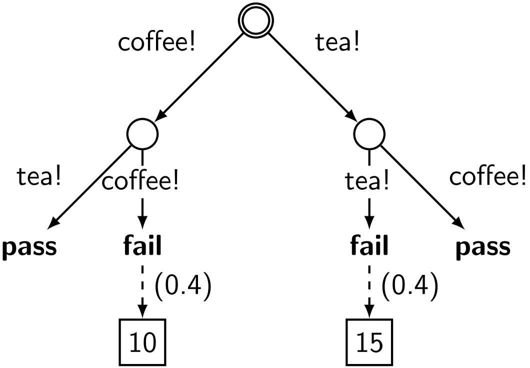


absCov	$\mathbb{P}$	$\times$
10	0.015	0.150
15	0.005	0.075
4	0.735	2.940
6	0.245	1.470 +

$$\mathbb{E}(absCov) = 4.635$$

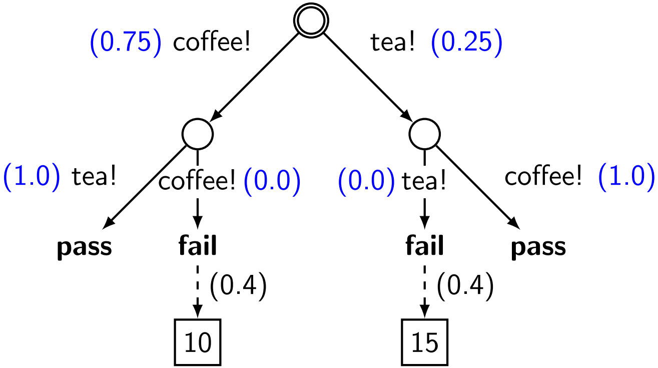
Branching probabilities

The *branching probabilities* p^{br} describe how the implementation is expected to behave



Branching probabilities

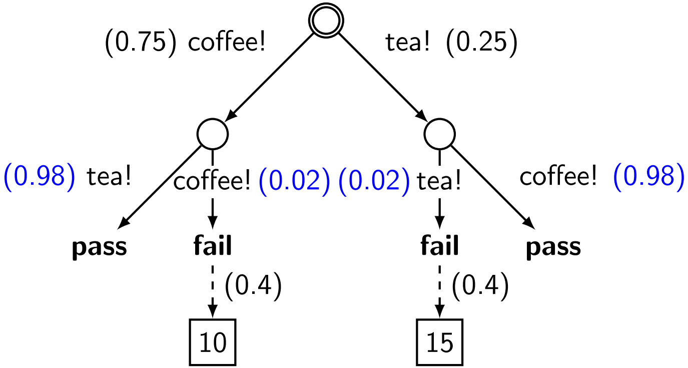
The *branching probabilities* p^{br} describe how the implementation is expected to behave



Branching probabilities

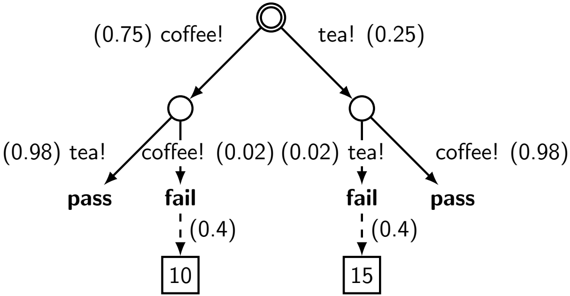
Branching probabilities

The *branching probabilities* p^{br} describe how the implementation is expected to behave



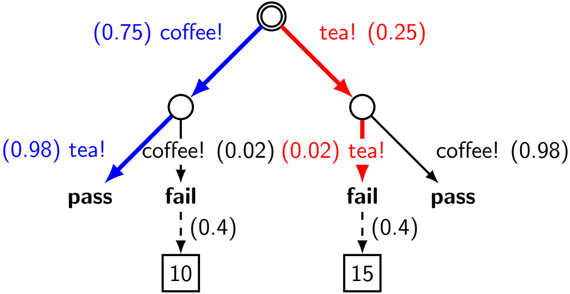
Sequences of executions

- Suppose we perform three executions of



Sequences of executions

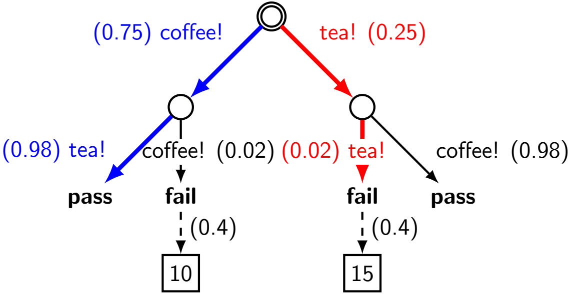
- Suppose we perform three executions of



- Possible observation: [blue, blue, red]

Sequences of executions

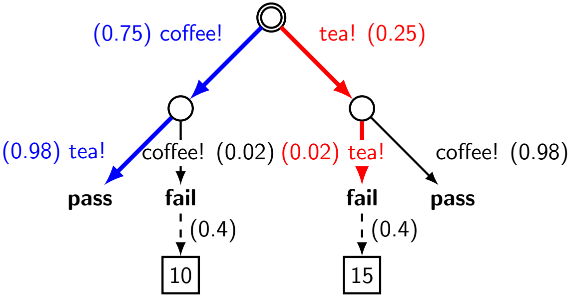
- Suppose we perform three executions of



- Possible observation: [blue, blue, red]
- Actual coverage: $15 + 6.4 = 21.4$

Sequences of executions

- Suppose we perform three executions of



- Possible observation: [blue, blue, red]
- Actual coverage: $15 + 6.4 = 21.4$
- Many observations possible: $O(|exec|^n)$

Expected actual coverage for a sequence of executions

Theorem

$$\mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \Bigg((1 - (1 - p^{\mathrm{to}}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{\mathrm{to}}(\sigma)^k (1 - p^{\mathrm{to}}(\sigma))^{n-k} \cdot (1 - p^{\mathrm{br}}(a \mid \sigma))^k \cdot (1 - (1 - p^{\mathrm{cbr}}(a \mid \sigma)^k)) \Bigg)$$

Expected actual coverage for a sequence of executions

Theorem

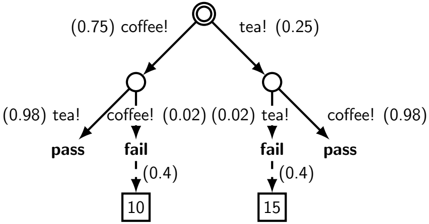
$$\mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \Bigg((1-(1-p^{\mathrm{to}}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{\mathrm{to}}(\sigma)^k (1-p^{\mathrm{to}}(\sigma))^{n-k} \cdot (1-p^{\mathrm{br}}(a \mid \sigma))^k \cdot (1-(1-p^{\mathrm{cbr}}(a \mid \sigma)^k)) \Bigg)$$

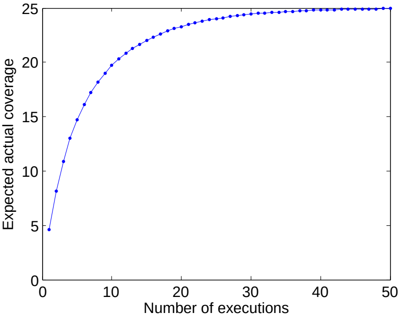
Expected actual coverage for a sequence of executions

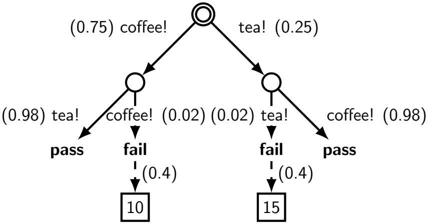
Theorem

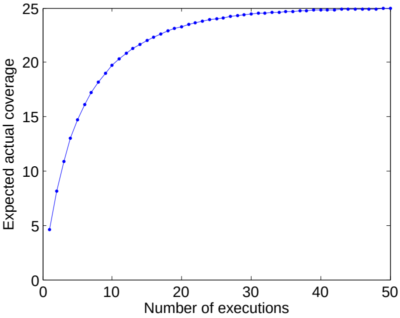
$$\mathbb{E}(actCov_n) = \sum_{\sigma a \in err_t} f(\sigma a) \cdot \Bigg((1 - (1 - p^{\mathrm{to}}(\sigma a))^n) \cdot 1 + \sum_{k=0}^{n} \binom{n}{k} p^{\mathrm{to}}(\sigma)^k (1 - p^{\mathrm{to}}(\sigma))^{n-k} \cdot (1 - p^{\mathrm{br}}(a \mid \sigma))^k \cdot (1 - (1 - p^{\mathrm{cbr}}(a \mid \sigma)^k)) \Bigg)$$

Asympotical behaviour of actual coverage









Theorem

$$\lim_{n\to\infty} \mathbb{E}(\text{actual coverage}_n) = \text{potential coverage}$$

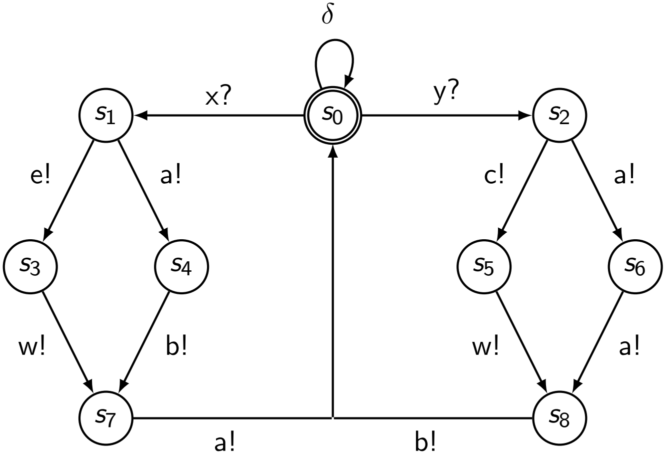
Actual coverage for test suites

- 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

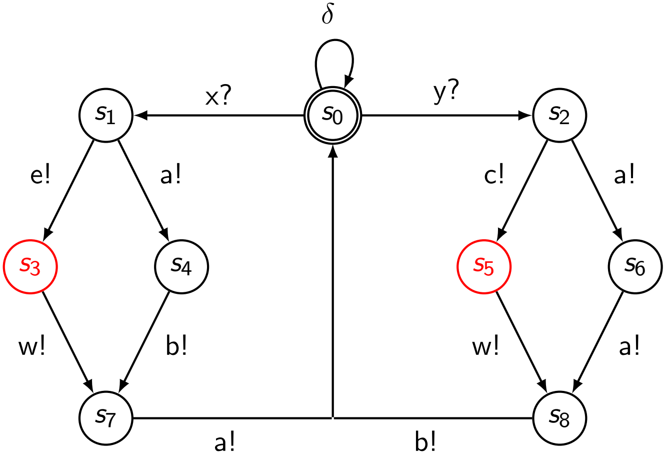
Theorem

$$\lim_{n \to \infty} \mathbb{E}(\text{actual coverage}_n) = \text{potential coverage}$$

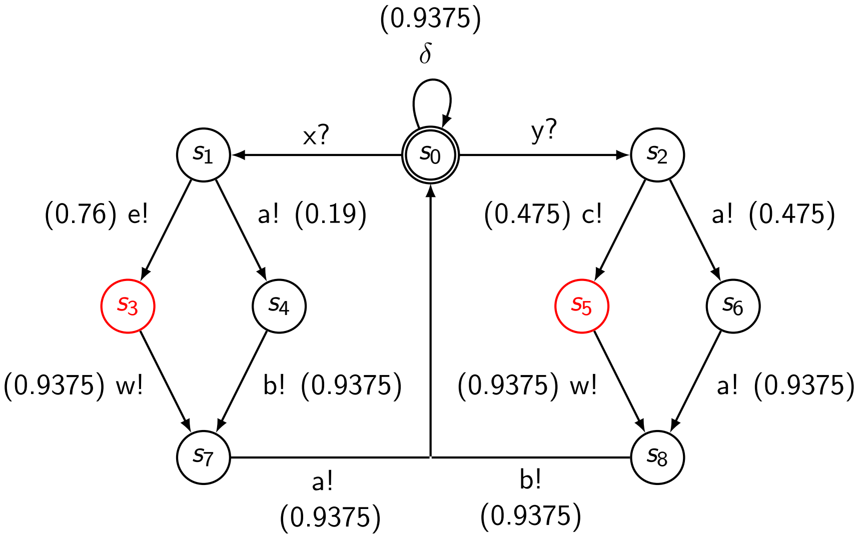
Example – chemical dispenser



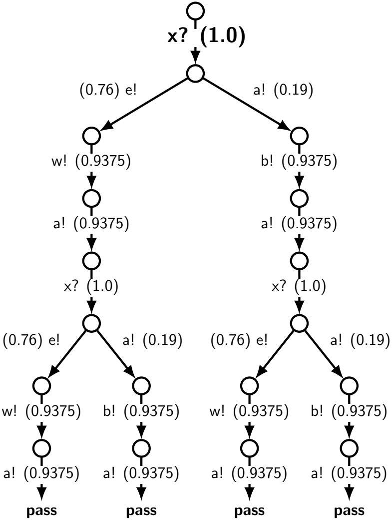
Example – chemical dispenser



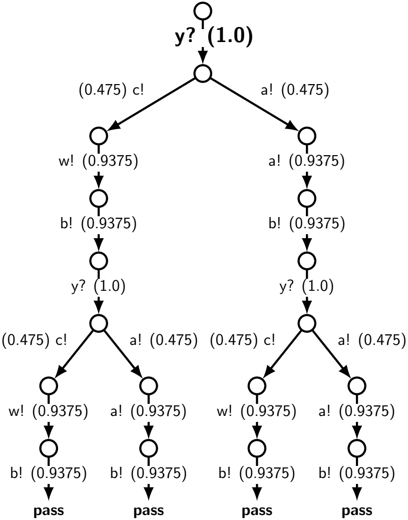
Example – chemical dispenser



Example – chemical dispenser



(a) Test case t_1 (pot.cov.: 1938)



(b) Test case t_2 (pot.cov.: 1938)

Prediction of actual coverage

$$
\begin{aligned}
\mathbb{E}(A^{1}_{t_1}) &= 197.0 \\
\mathbb{E}(A^{5}_{t_1}) &= 729.1 \\
\mathbb{E}(A^{10}_{t_1}) &= 1076.9 \\
\mathbb{E}(A^{50}_{t_1}) &= 1704.6 \\
\mathbb{E}(A^{250}_{t_1}) &= 1917.7
\end{aligned}
\qquad
\begin{aligned}
\mathbb{E}(A^{1}_{t_2}) &= 156.8 \\
\mathbb{E}(A^{5}_{t_2}) &= 639.8 \\
\mathbb{E}(A^{10}_{t_2}) &= 1032.1 \\
\mathbb{E}(A^{50}_{t_2}) &= 1848.0 \\
\mathbb{E}(A^{250}_{t_2}) &= 1938.0
\end{aligned}
$$

Prediction of actual coverage

$$
\begin{array}{lclclcl}
\mathbb{E}(A^{1}_{t_1}) & = & 197.0 & > & \mathbb{E}(A^{1}_{t_2}) & = & 156.8 \\
\mathbb{E}(A^{5}_{t_1}) & = & 729.1 & > & \mathbb{E}(A^{5}_{t_2}) & = & 639.8 \\
\mathbb{E}(A^{10}_{t_1}) & = & 1076.9 & > & \mathbb{E}(A^{10}_{t_2}) & = & 1032.1 \\
\mathbb{E}(A^{50}_{t_1}) & = & 1704.6 & < & \mathbb{E}(A^{50}_{t_2}) & = & 1848.0 \\
\mathbb{E}(A^{250}_{t_1}) & = & 1917.7 & < & \mathbb{E}(A^{250}_{t_2}) & = & 1938.0
\end{array}
$$

Simulation and evaluation actual coverage

	t_1			t_2		
n	$\mathbb{E}(A^n_{t_1})$	$\overline{\text{Sim.}}$	std.	$\mathbb{E}(A^n_{t_2})$	$\overline{\text{Sim.}}$	std.
1	197.0	213.3	50.1	156.8	155.1	60.8
5	729.1	762.1	84.0	639.8	629.7	107.0
10	1076.9	1112.6	104.8	1032.1	1013.3	114.8
50	1704.6	1743.3	62.4	1848.0	1831.2	39.5
250	1917.7	1925.8	11.2	1938.0	1938.0	0.0

	t_1			t_2		
n	$\mathbb{E}(A^n_{t_1})$	$\overline{\text{Sim.}}$	std.	$\mathbb{E}(A^n_{t_2})$	$\overline{\text{Sim.}}$	std.
1	197.0	229.1	48.4	156.8	174.7	52.0
5	729.1	813.9	56.5	639.8	711.6	89.4
10	1076.9	1167.9	94.8	1032.1	1133.6	92.9
50	1704.6	1757.3	62.9	1848.0	1890.7	19.4
250	1917.7	1926.0	11.1	1938.0	1938.0	0.0

Main results

- 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

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

For more details, see my Master's Thesis
(`wwwhome.cs.utwente.nl/~timmer`)

Conclusions and future work

Main results

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

For more details, see my Master's Thesis
(`wwwhome.cs.utwente.nl/~timmer`)

Directions for future work

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

Questions