Mutation Score, Coverage, Model Inference: Quality Assessment for *t*-way Combinatorial Test-Suites

Hermann Felbinger, Franz Wotawa, Mihai Nica
Graz University of Technology

Motivation

- Extend existing empirical evaluation results
- Evaluate new quality assessment method

Assessment Methods

- Mutation score
- Code coverage
- Model inference based approach

Mutation Score

- Create mutant by modifying original program under test
- Source code or binary
- At least one test in test-suite yields different verdict (fail/pass) when executing original program and mutant -> mutant killed
- Very expensive method
- Mutation framework Major[1]

[1] http://mutation-testing.org/

Code Coverage

- Instruction, branch, MC/DC, …
- Source code or binary
- May be intrusive
- Source code coverage tool CodeCover[1]

[1] http://codecover.org/

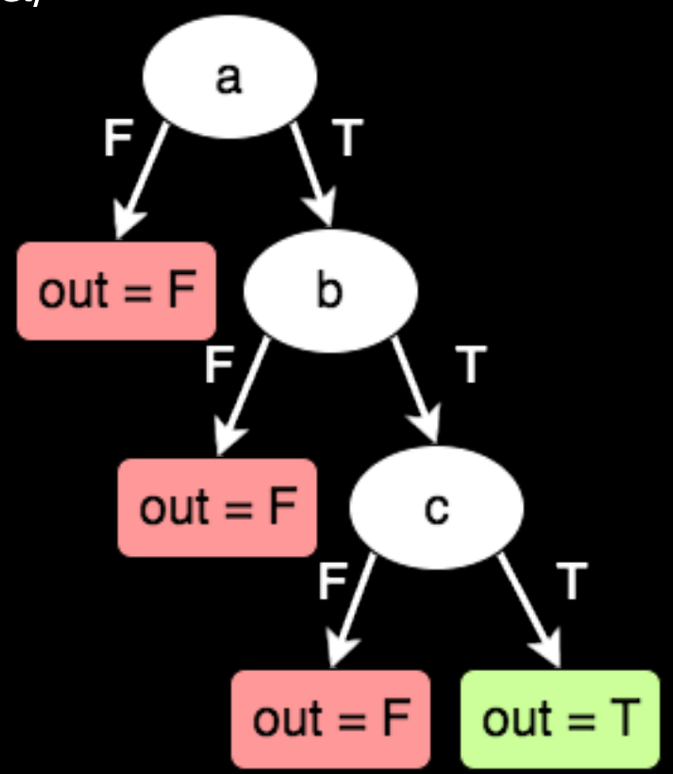
Model Inference 1/2

- As model infer a decision tree from a test-suite
- C4.5 algorithm to create decision tree
- C4.5 is based on entropy and information gain
- Implementation in Weka[1] called J48

[1] http://www.cs.waikato.ac.nz/ml/weka/

Test-Suite

	a	b	c	out
1	T	T	T	T
2	F	T	T	F
3	T	F	T	F
4	T	T	F	F



Model Inference 2/2

- Assume a test-suite $TS_{t_{max}}$ to be of high quality
- Assess quality by comparing a test-suite TS to $TS_{t_{max}}$
- TS is of high quality if
  1. The inferred model contains all outcomes of the set of possible outcomes O
  2. The inferred model classifies a set of test-data TD correctly to these leaf nodes

$$TD = TS_{t_{max}} \setminus \bigcup_{t=1}^{t<t_{max}} TS_t$$

Model Inference Based Test-Suite Quality Assessment 1/2

- For a test-suite TS - depends on:
  1. $RMSE_{TS}$ of the inferred model
  2. RMSE after classifying TD

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$$

$p_1, .., p_n$ are the outcomes of the inferred model

$a_1, .., a_n$ are the reference outcomes

3. The difference of the number of outcomes L that are in the inferred model and O

$$MC = \frac{|O| - |L|}{|O|}$$

Model Inference Based Test-Suite Quality Assessment 2/2

$$MI = 1 - (RMSE - MC + RMSE_{TS})$$

Research Questions

1. How does incrementing *t* affect the test-suite quality?
2. Does a model inference based test-suite quality assessment approach show similar differences for test-suite quality of test-suites generated with different *t*, as mutation score or code coverage?

Example Programs

name	SLOC	#mutants
BMI	19	28
Triangle	30	35
UTF8	56	147
TCAS	100	41
J48	3406	3107
Soot-PDG	1701	567

Test-suite Generation

- Generated t-way combinatorial test-suites using ACTS 3.0[1]

[1]http://csrc.nist.gov/groups/SNS/acts/index.html

Input Models & Constraints 1/3

BMI Input Model

Parameter	Values
height	{1.6, 1.8, 2.0, 2.2}
weight	{73, 74, 99, 100, 119, 120, 159, 160}

Triangle Input Model

Parameter	Values
a	{-1, 0, 1, 3, 4, 5, 2^{31}-1}
b	{-1, 0, 1, 3, 4, 5, 2^{31}-1}
c	{-1, 0, 1, 3, 4, 5, 2^{31}-1}

UTF8 Input Model

Parameter	Values
b1	{0, -1, 127, -128, -62, -63, -33, -32, -31, -30, -20, -19, -18, -17, -16, -15, -14, -13, -12, -11}
b2	{-128, -65, -64, -97, -96, -112, -113, ?}
b3	{-128, -65, -64, ?}
b4	{-128, -65, -64, ?}

Constraints of UTF8 Example

(b2 == ?) => (b3 == ?)
(b3 == ?) => (b4 == ?)

Input Models & Constraints 2/3

TCAS Input Model

Parameter	Values
Cur_Vertical_Sep	{299, 300, 601}
High_Confidence	{0, 1}
Two_of_Three_Reports_Valid	{0, 1}
Own_Tracked_Alt	{1, 2}
Own_Tracked_Alt_Rate	{600, 601}
Other_Tracked_Alt	{1, 2}
Alt_Layer_Value	{0, 1, 2, 3}
Up_Separation	{0, 399, 400, 499, 500, 639, 640, 739, 740, 840}
Down_Separation	{0, 399, 400, 499, 500, 639, 640, 739, 740, 840}
Other_RAC	{0, 1, 2}
Other_Capability	{1, 2}
Climb_Inhibit	{0, 1}

J48 Input Model

Parameter	Values
-U	{F, T}
-O	{F, T}
-C	{0.0, 0.1, 0,9, 1.0}
-M	{0, 1, 2, 5}
-R	{F, T}
-N	{0, 3, 10}
-dNMSAV	{F, T}
-S	{F, T}
-L	{F, T}
-A	{F, T}
-J	{F, T}
-Q	{0, 1, 100}
-B	{F, T}

Constraints of J48 Example

!(U & S)
!(U & R)
!R | !C
!U | !C
R | !N

TU Graz

Input Models & Constraints 3/3

Soot-PDG Control Statements

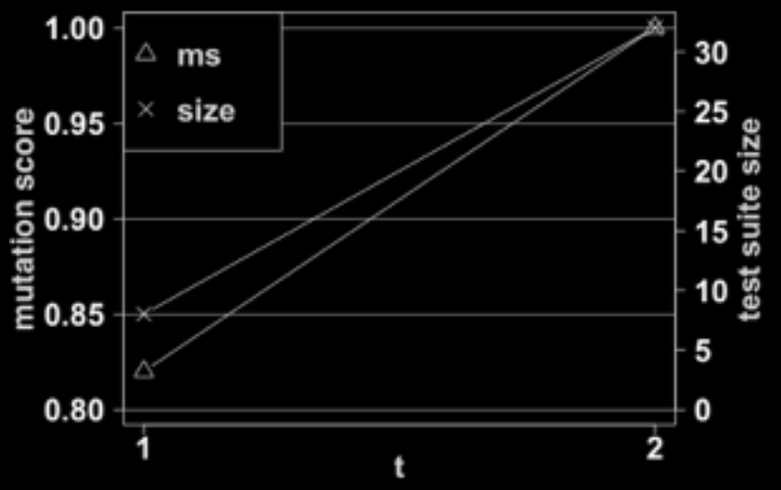
Group 1	Group 2	Group 3
IF-ELSE_IF-ELSE	ENHANCED_FOR	THROW
SWITCH	ENHANCED_FOR_BREAK	RETURN
SWITCH_BREAK	ENHANCED_FOR_CONTINUE	CALLABLE
TRY_CATCH_FINALLY	BASIC_FOR	NOP
LINEAR_RECURSION	BASIC_FOR_BREAK	
NOP	BASIC_FOR_CONTINUE	
	WHILE	
	WHILE_BREAK	
	WHILE_CONTINUE	
	DO_WHILE	
	DO_WHILE_BREAK	
	DO_WHILE_CONTINUE	
	NOP	

```
public class PDGInput {
    public void run int var, int[] array) {
        for (var = 0; var < 10; var++) {
            while (var > 0) {
                for (int e0 : array) {
                    for (var = 0; var < 10; var++) {
                        for (int e1 : array) {
                            System.out.println "NOP " + var);
                            System.out.println "ENHANCED_FOR " + var);
                        }
                        break;
                    }
                …
```

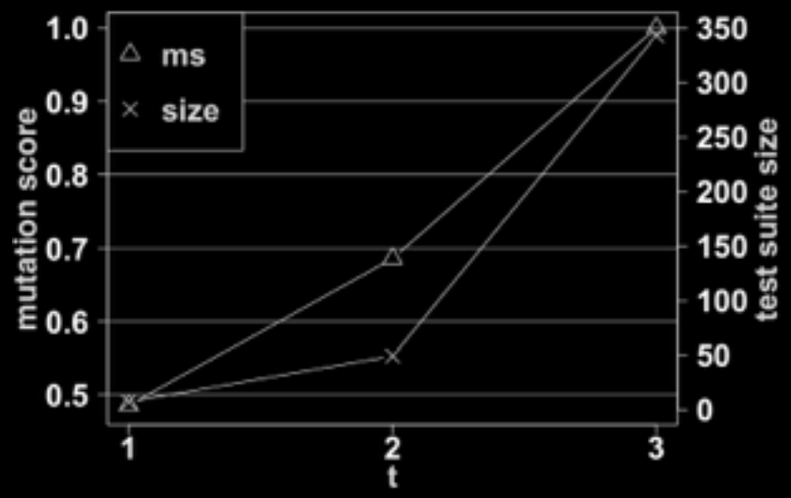
Soot-PDG Input Model

Parameter	Values
L1	{1, 2}
L2	{1, 2}
L3	{1, 2}
L4	{1, 2}
L5	{1, 2}
L6	{1, 2, 3}

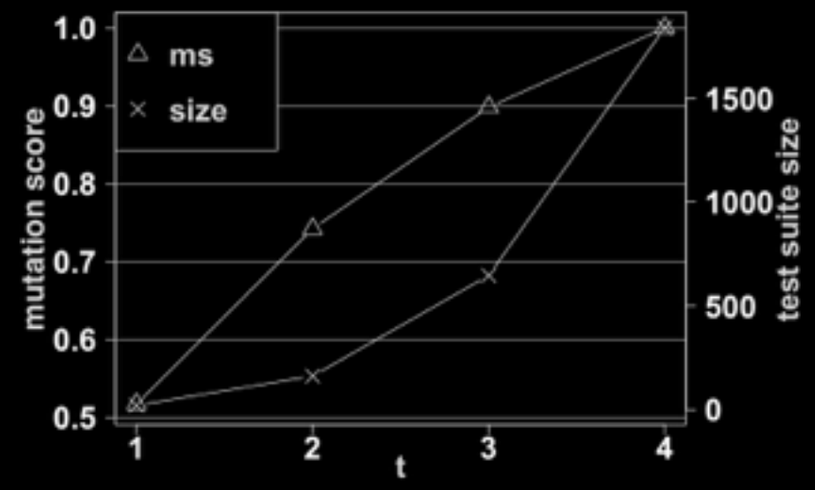
Mutation Score Results



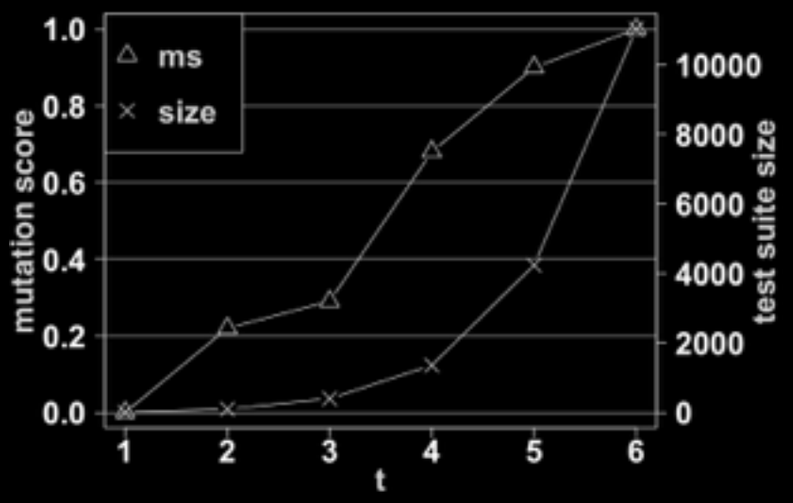
BMI



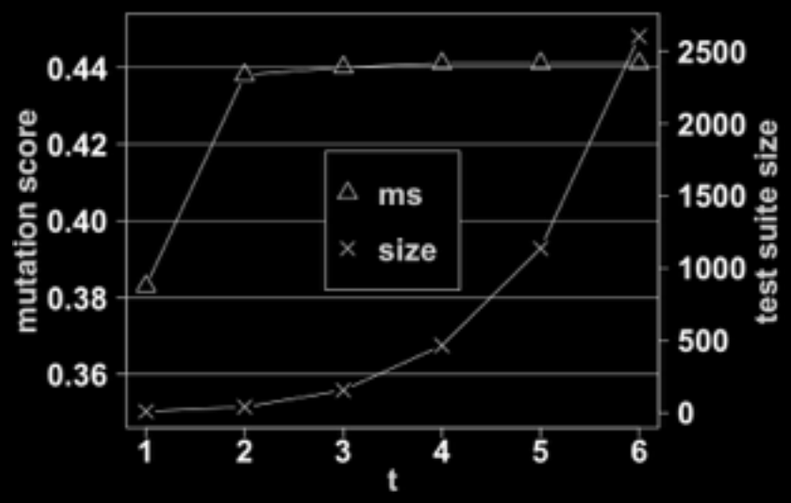
Triangle



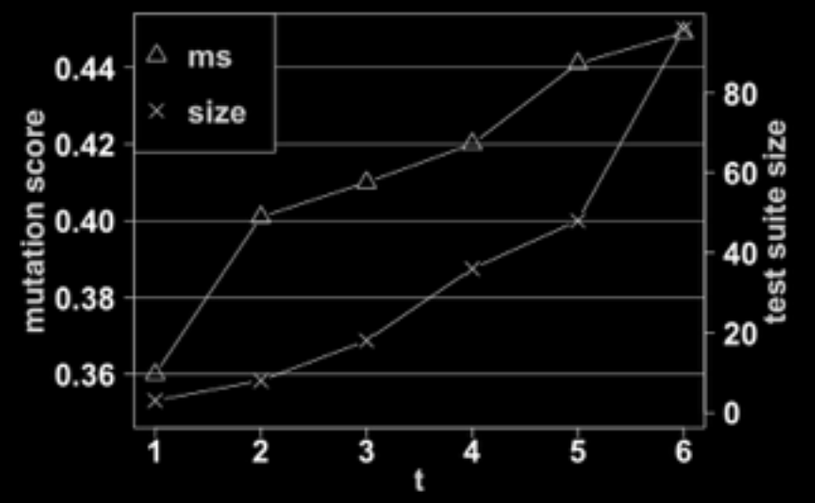
UTF8



TCAS



J48



Soot-PDG

Code Coverage Results

		t					
	coverage	1	2	3	4	5	6
BMI	statem.	85.71	100.00				
	branch	87.50	100.00				
	MC/DC	87.50	100.00				
Triangle	statem.	61.54	92.31	100.00			
	branch	75.00	91.67	100.00			
	MC/DC	56.25	75.00	100.00			
UTF8	statem.	85.71	100.00	100.00	100.00		
	branch	85.00	100.00	100.00	100.00		
	MC/DC	57.50	100.00	100.00	100.00		
TCAS	statem.	50.00	94.44	94.44	97.22	97.22	97.22
	branch	08.33	83.33	83.33	91.67	91.67	91.67
	MC/DC	15.00	70.00	70.00	85.00	85.00	85.00
J48	statem.	48.22	49.81	49.81	49.81	49.81	49.81
	branch	48.06	51.74	51.74	51.74	51.74	51.74
	MC/DC	48.24	50.88	50.88	50.88	50.88	50.88
Soot-PDG	statem.	61.95	66.36	68.48	69.77	72.36	73.11
	branch	37.66	44.69	47.77	48.44	51.84	52.93
	MC/DC	43.06	49.55	52.20	52.38	55.24	56.15

Model Inference Results

	BMI	Triangle	UTF8	TCAS	J48	Soot-PDG
|O|	5	4	2	3	161	125
|TD|	24	288	1115	10860	2633	96

MI Results

t	BMI	Triangle	UTF8	TCAS	J48	Soot-PDG
1	0.2836	0.0512	0.5558	0.2792	0	0
2		0.43	0.7913	0.4978	0.0132	0
3			0.9088	0.5875	0.2780	0
4				0.9093	0.6808	0.0617
5				0.9595	0.8604	0.1293

Conclusions

- The quality of *t*-way combinatorial test-suites increases with higher strength
- MI is only applicable under restricted conditions
- For test-suites with |O| < |TS|, the results of mutation score, coverage and MI are similar
- MI calculation is very fast and not intrusive
- Extension of empirical evaluation for MI necessary
- Investigation of MI based reduction approach