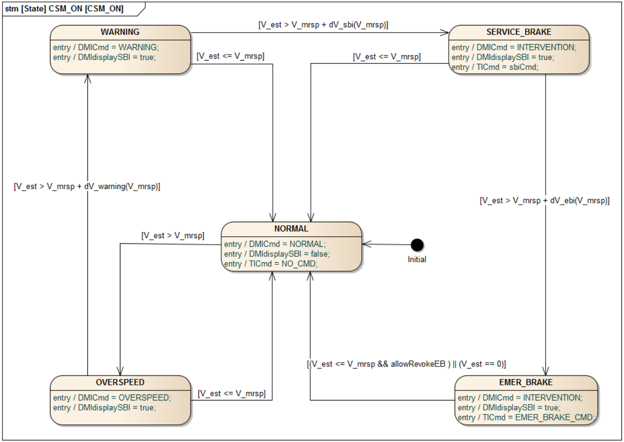
Experimental Evaluation of a Novel Equivalence Class Partition Testing Strategy

Felix Hübner Wen-ling Huang Jan Peleska

Department of Mathematics and Computer Science
University of Bremen, Germany
http://informatik.uni-bremen.de/agbs

22nd July 2015

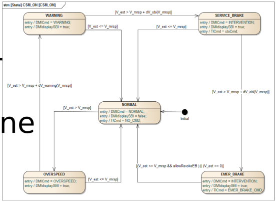
Safety in the railway domain



ETCS (European Train Control System)

Test Case Generation

SysML state machine



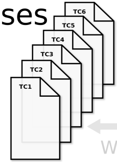
State-Transition-System

I/O Equivalence Class Abstraction

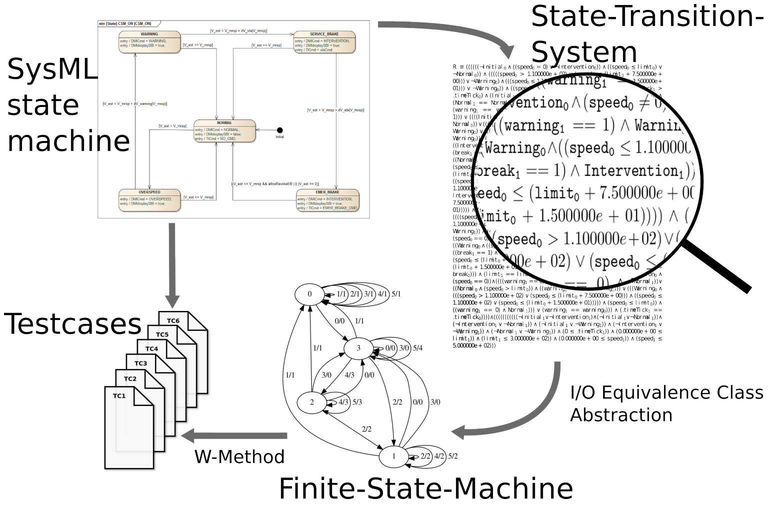
Finite-State-Machine

W-Method

Testcases



Test Case Generation



Basics

Strategy works on semantic level: deterministic **Reactive State Transition System (RSTS)**

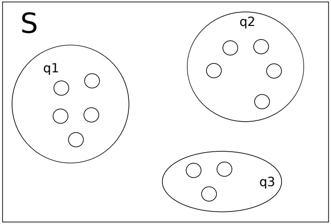
$$\mathcal{S} = (S, s_0, R) \tag{1}$$
$$S \subseteq V \to D \text{ variable valuation functions} \tag{2}$$
$$V = I \cup M \cup O \text{ input, internal, output variable symbols} \tag{3}$$
$$D = \text{ variable domains} \tag{4}$$
$$R = S \times S \text{ transition relation} \tag{5}$$

Types of input variables may be infinite.
Types of internal and output variables have to be finite.

Basics

I/O equivalence

- Two states s and s' are equivalent, if every input trace applied to these states lead to the same observable output trace
  $s \sim s' \equiv \forall \iota = \vec{c}_1 \dots \vec{c}_n \in D_I^* : (s/\iota)|_O = (s'/\iota)|_O$
- factorise states into I/O-equivalence classes q

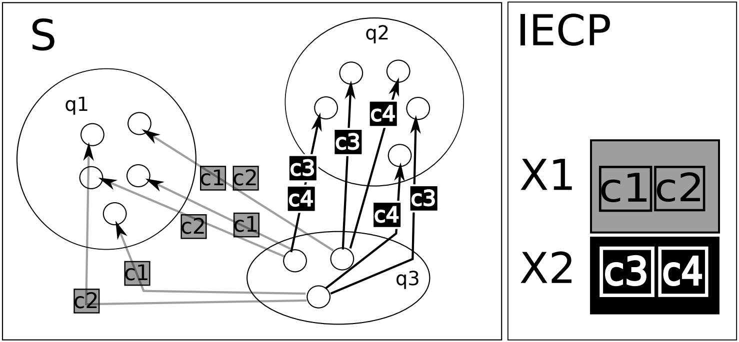


- **number of I/O-equivalence classes is finite**
- Two systems are equivalent, if their initial states are I/O equivalent

Basics

Input Equivalence Class Partitioning (IECP)

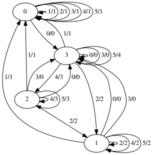
- factorise input space into a **finite** number of input equivalence classes (IEC) X, such that, inputs c_1 and c_2 are equivalent ($c_1 \sim_I c_2$), if:
  $\forall q \forall s \in q \exists q' : (s/c_1) \in q' \Leftrightarrow (s/c_2) \in q' \wedge (s/c_1)|_O = (s/c_2)|_O$



Basics

DFSM abstraction

- I/O equivalence class factorisation and IECP induce complete DFSM abstraction of the test model
- concrete alphabet can be extracted from IECP
- complete DFSM strategy can be applied, e.g. W-Method



Completeness of Equivalence Class Partitioning Strategy

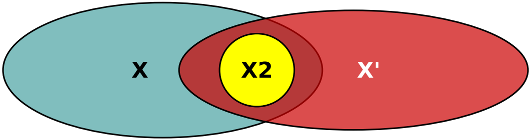
Equivalence class testing method is **complete** with respect to a given **fault domain**

- **soundness**: every correct behaviour of an SUT will be accepted
- **exhaustiveness**: every erroneous behaviour of the SUT will be rejected, provided that the true SUT behaviour is inside a (very large) set of pre-defined behavioural models, the **fault domain**
- experiments have shown that the equivalence class strategy also shows superior test strength for SUT behaviours **not restricted** to the fault domain

Completeness of Equivalence Class Partitioning Strategy

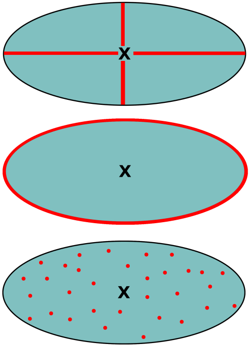
Faultdomain $\mathcal{D}(\mathcal{S}, m, \mathcal{I}_2)$ contains all systems $\mathcal{S}'$

- the number of I/O-equivalence classes of $\mathcal{S}'$ is less or equal m
- for every input equivalence class X of $\mathcal{S}$ and every input equivalence class X' of $\mathcal{S}'$: $X \cap X' \neq \emptyset \Rightarrow \exists X_2 \in \mathcal{I}_2 \subseteq X \cap X'$



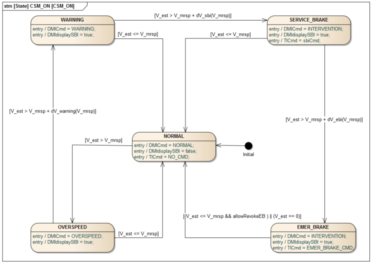
Heuristics

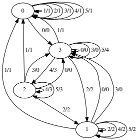
- MCDC
- boundary value tests
- combination with random testing



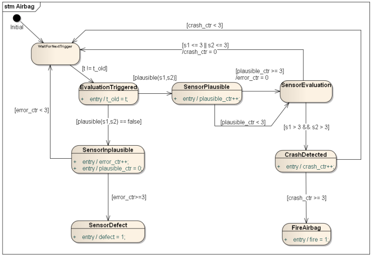
Experimental Evaluation – Models

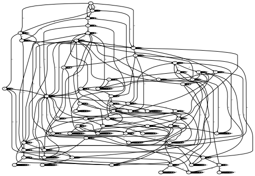
Ceiling Speed Monitor





Airbag Controller





Experimental Evaluation

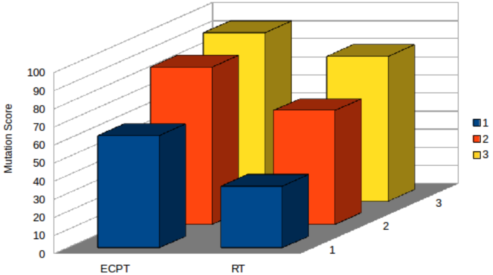
- evaluate test strength of generated test suite for SUTs **not restricted** to the faultdomain
- compare mutation score to random testing
- create a correct implementation of the test model (Java)
- build mutants using the μJava mutation tool

```
public class CeilingSpeedMonitor extends SUT
{

   public double V_mrsp = 0.0;

   public double V_est = 0.0;

    public int b = 0;

    public  void update()
    {
       if (V_est != V_mrsp) {
          b = 1;
       }
       if (V_est == 0) {
          b = 0;
```

- $\text{mutation score} = \frac{\text{number of mutants that did not pass the test suite}}{\text{number of mutants}}$

Experimental Evaluation – Results

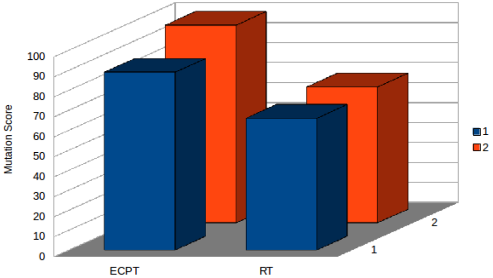
Ceiling Speed Monitor



			Mutation Score	
		No. testcases	ECPT	RT
1	no heuristics	21	62 %	34 %
2	MCDC	186	87 %	64 %
3	boundary values	610	93 %	80 %

Experimental Evaluation – Results

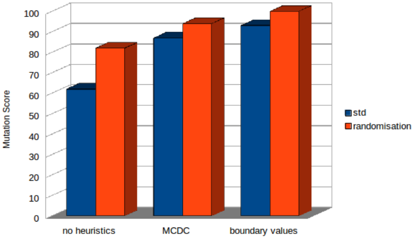
Airbag Controller



			Mutation Score	
		No. testcases	ECPT	RT
1	no heuristics	368	89 %	66 %
2	boundary values	3248	99 %	68 %

Experimental Evaluation – Combination with Random Testing

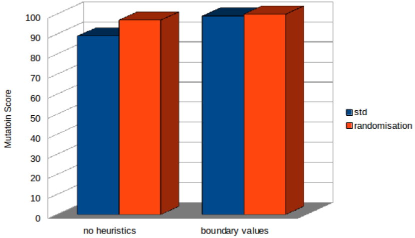
Ceiling Speed Monitor



	Mutation Score std	Mutation Score randomisation	Δ
no heuristics	62 %	82 %	+ 20 %
MCDC	87 %	94 %	+ 7 %
boundary values	93 %	100 %	+ 7 %

Experimental Evaluation – Combination with Random Testing

Airbag Controller



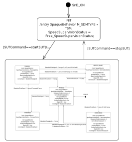
	Mutation Score		Δ
	std	randomisation	
no heuristics	89 %	97 %	+ 8 %
boundary values	99 %	100 %	+ 1 %

Conclusion

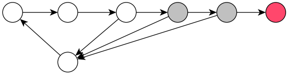
- experimental evaluation of a complete equivalence class testing strategy
- results apply for SUTs **not restricted** to the fault domain (completeness assumption does not apply, approx. 50 % of the mutants outside the fault domain)
- ECPT has significantly greater strength than conventional RT
- heuristics and randomisation of equivalence classes increase the test strength even further

Future and Ongoing Work

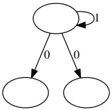
- Evaluation and Improvement of ETCS Test Cases for the Ceiling Speed Monitor



- evaluation of the effect of a "randomisation in the *m*-dimension"



- generalisation of the ECPT for non-deterministic models



Thank you for your attention!

I will now answer remaining questions.