Markus Wagner

Maximising Axiomatization Coverage and Minimizing Regression Testing Time

Joint work with Bernhard Beckert (KIT), Thorsten Bormer (KIT), and Mahmoud Bokhari (UoA)

Who guards the guardians?

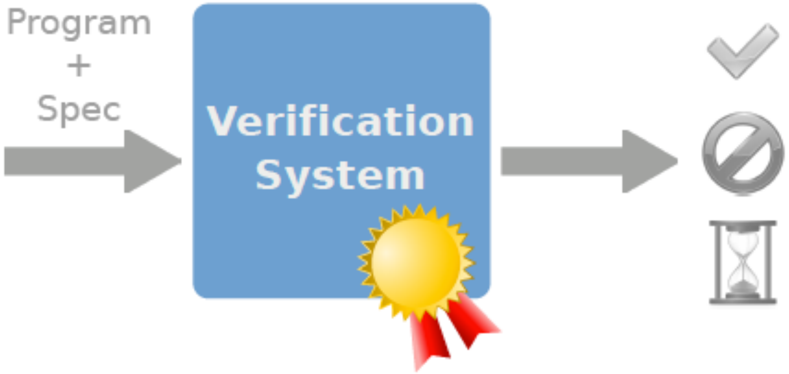
How to improve trust in **formal verification systems**?

$$a = b \vdash 2 = 1$$

Modern verification systems are large and complex systems

- Soundness bugs are not rare
- Such bugs are often hard to detect in a real proof

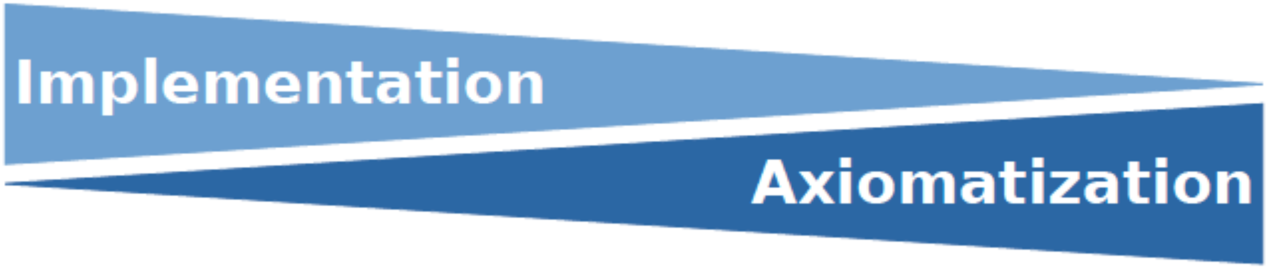
“Auto-active” Verification Systems



Validating verification systems by

- Formal methods
- Code inspection
- Testing
- …

Program Language Semantics



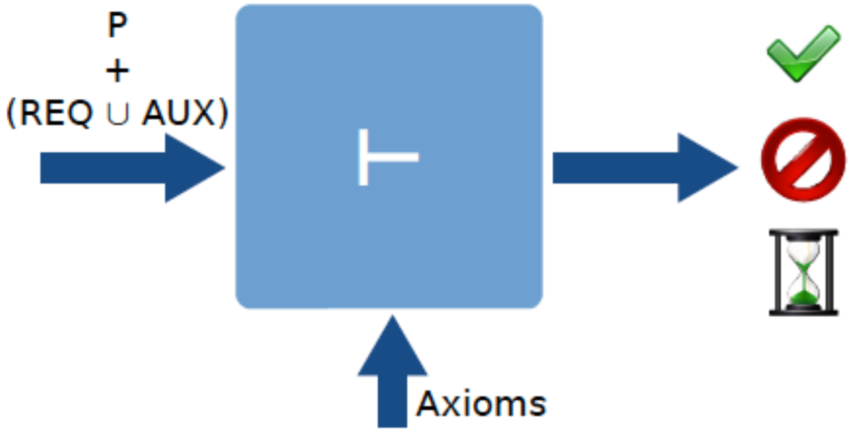
Static checkers Verifying compilers Logic frameworks

We have to test both!

But how to determine the quality of the test cases?*

*work started in 2008

Test Cases

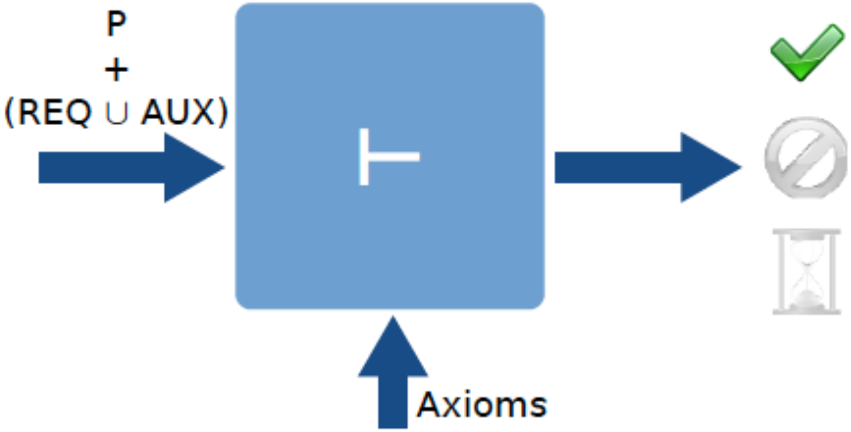


A test case is a program *P*, together with *REQuirements* and *AUXiliary specifications*.

Manually creating test cases is extremely time-consuming.

Computing coverage for the test cases takes from a few minutes to several hours.

Completeness Coverage

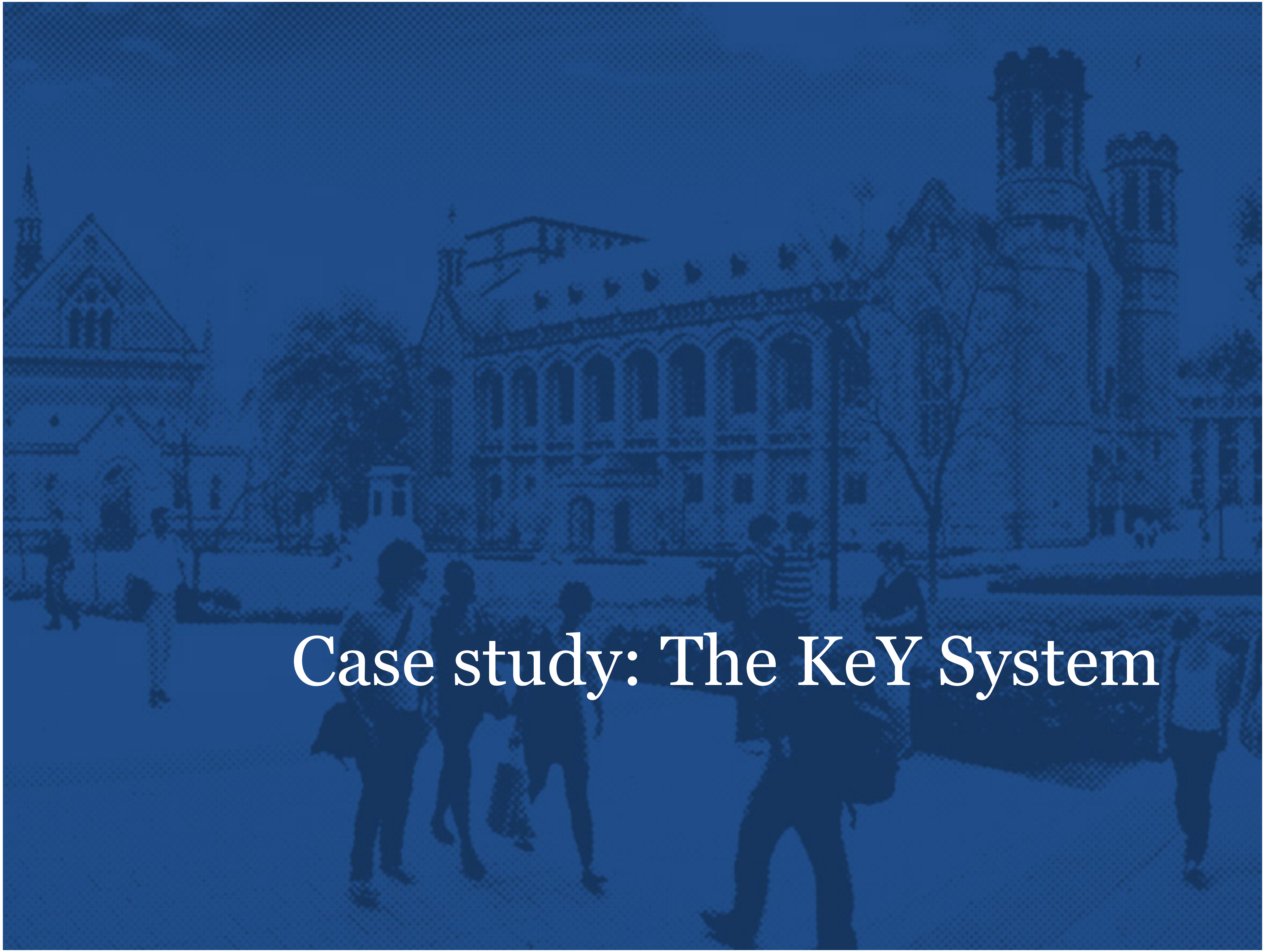


Definition (Completeness Coverage, TAP 2013)

A test case $P + (\, REQ \cup AUX\,)$ covers the set of *Axioms* if

- $Axioms \vdash P + (\, REQ \cup AUX\,)$
- and this does not hold for $Axioms' \subsetneq Axioms$

Note: covered set *Axioms* is not uniquely defined by the test case



Case study: The KeY System

The KeY System

Karlsruhe Institute of Technology
Technische Universität Darmstadt
Chalmers University of Technology

- Deductive verification system for JavaCard
- Sequent calculus for Java Dynamic Logic, uses symbolic execution for Java programs
- Interactive verification with automatic proof mode

Important

- The semantics of JavaCard is encoded in 1520 axioms (“small, well-understood set of sentences”)

Coverage Example: PostConditionTaclets2

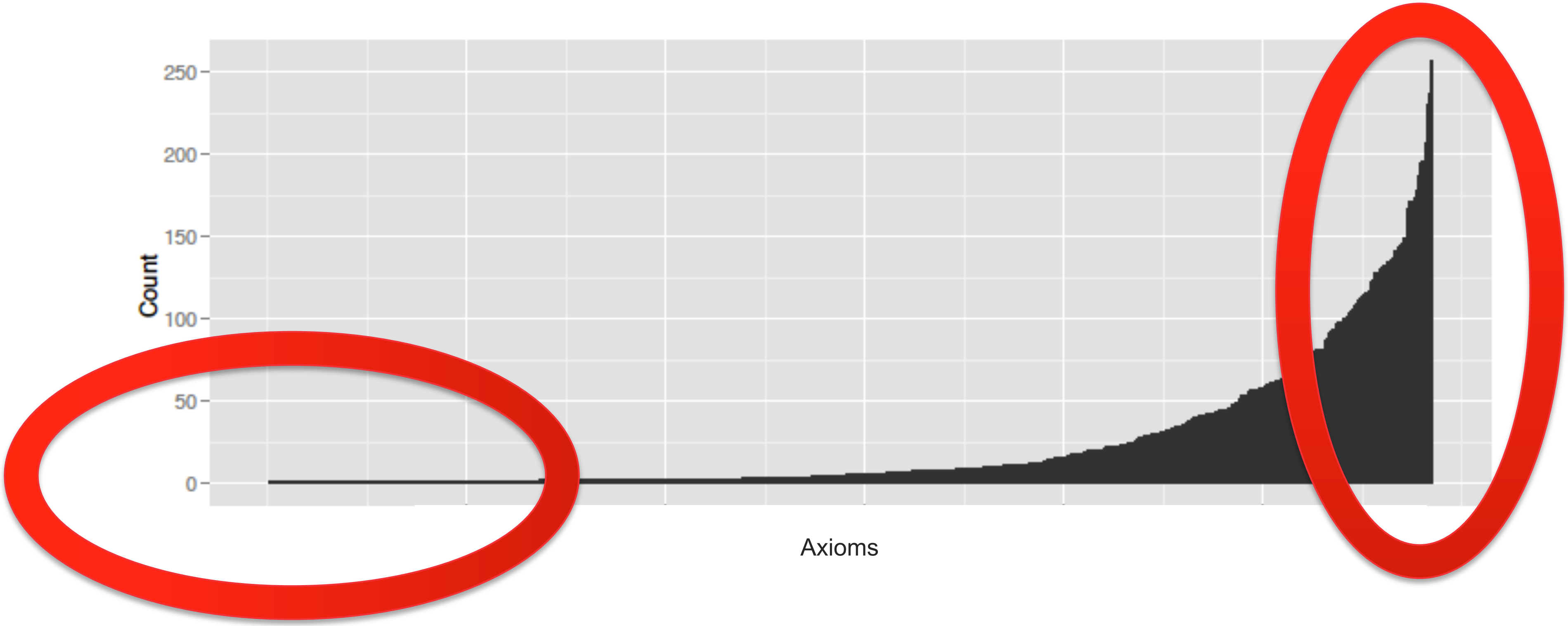
- Code Coverage (EMMA Tool)

Name	Class %	Method %	Line %
Coverage	86% (1.175 out of 1.361)	43% (7.369 out of 17.260)	35% (31.873 out of 92.139)

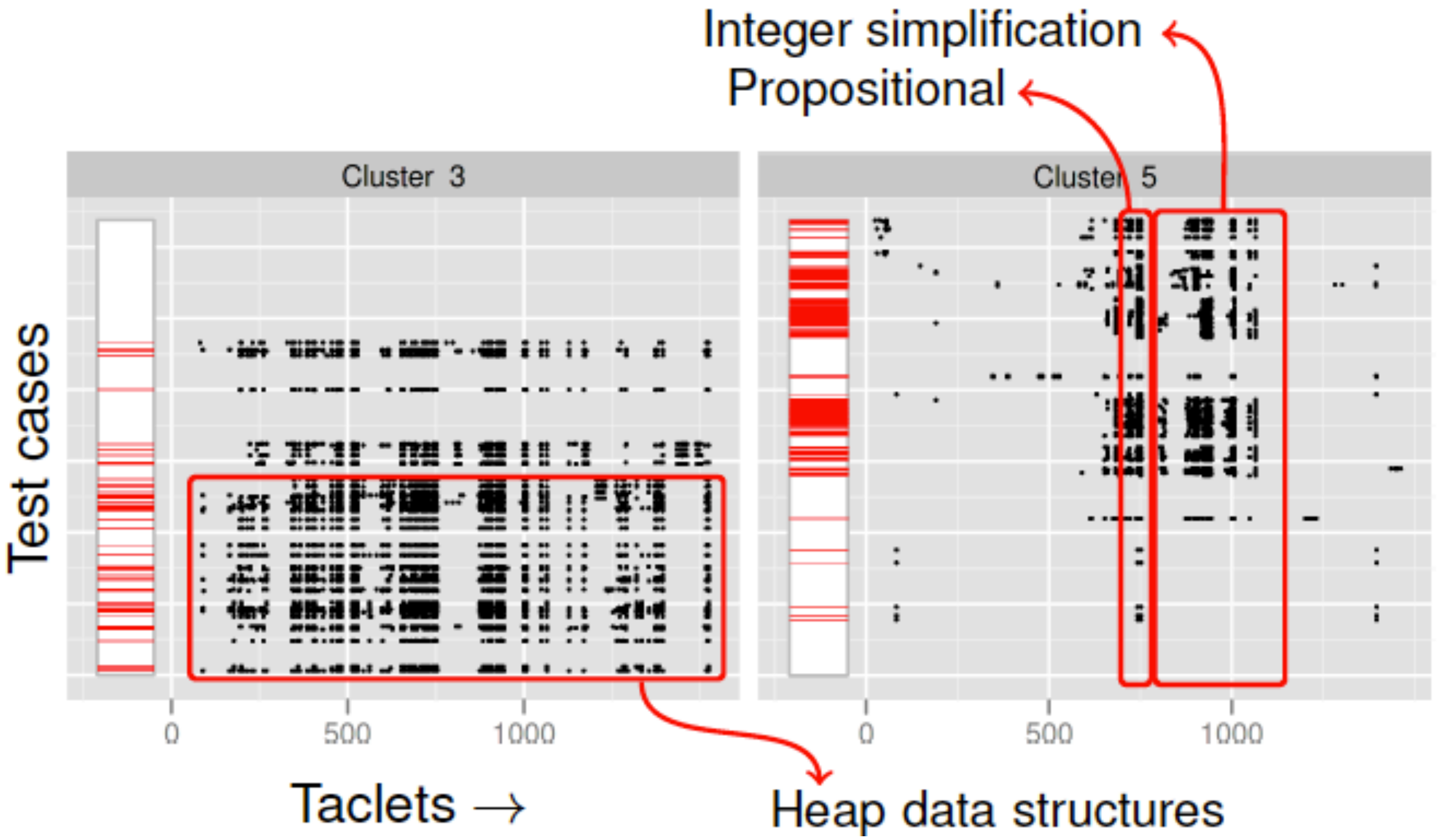
- Axiom Coverage: 0.32% (5 out of 1520)

Coverage Results (naïve, TAP 2013)

The 319 completeness tests of KeY covered 40% of all axioms (611 out of 1520).

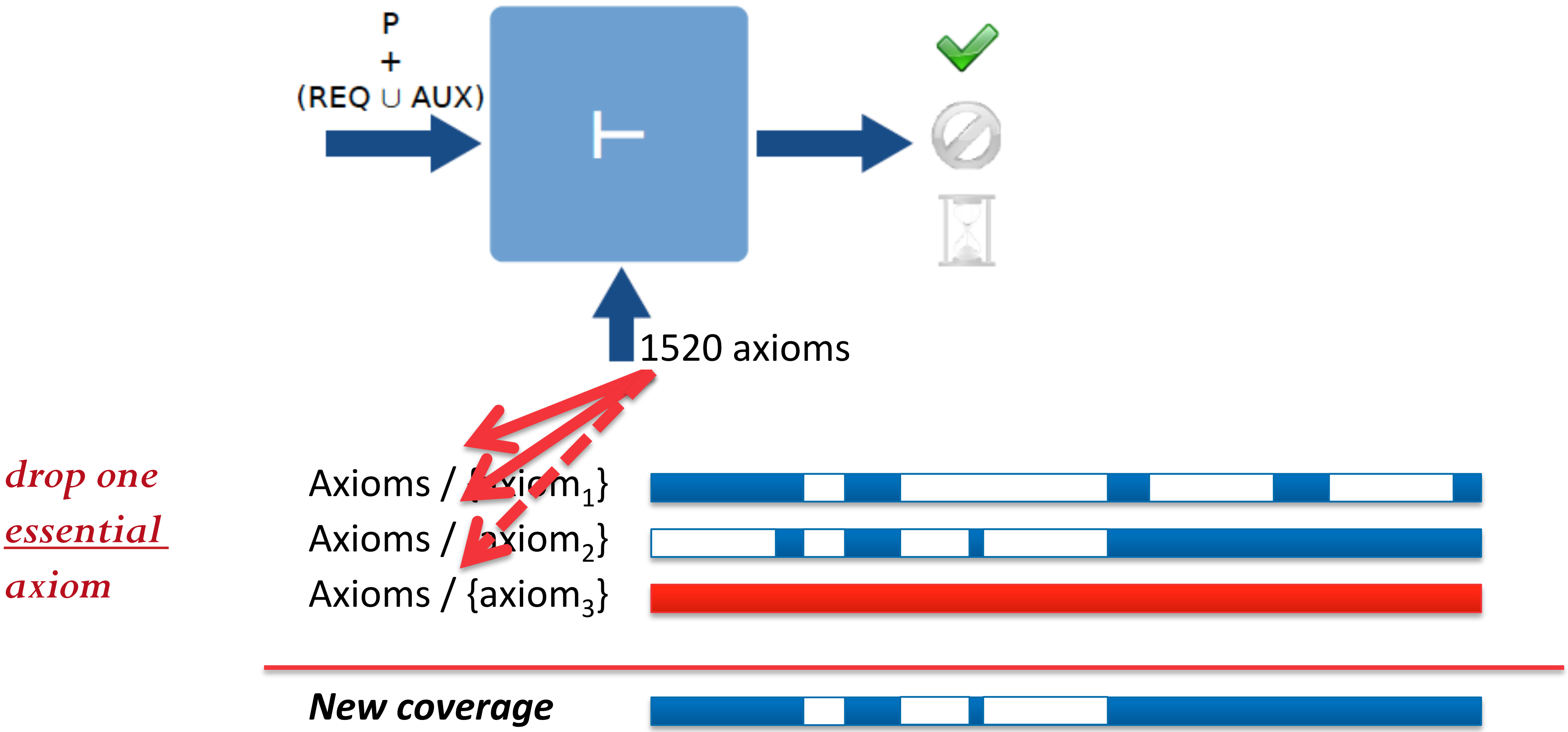


What is tested?



Heuristic Approaches

Reusing Test Cases



P + (REQ ∪ AUX)

$\vdash$

1520 axioms

drop one essential axiom

Axioms / {axiom$_1$}

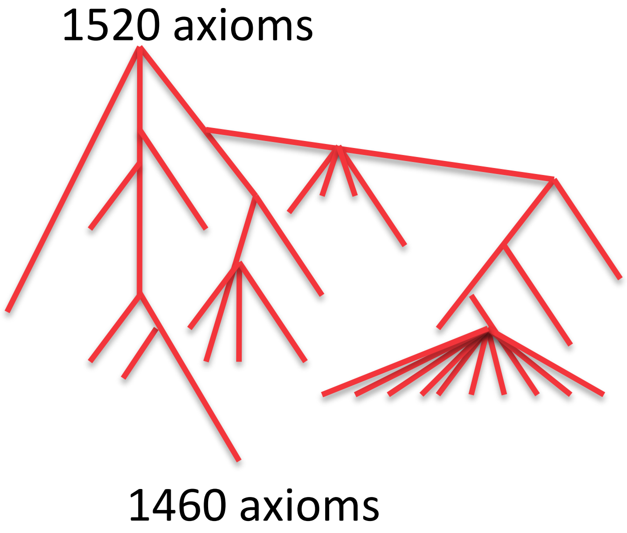
Axioms / {axiom$_2$}

Axioms / {axiom$_3$}

New coverage

Idea: given a test case *T*, run the tool with just a subset of the 1520 axioms.

Reusing Test Cases



Note:

- 24h per heuristic per test case

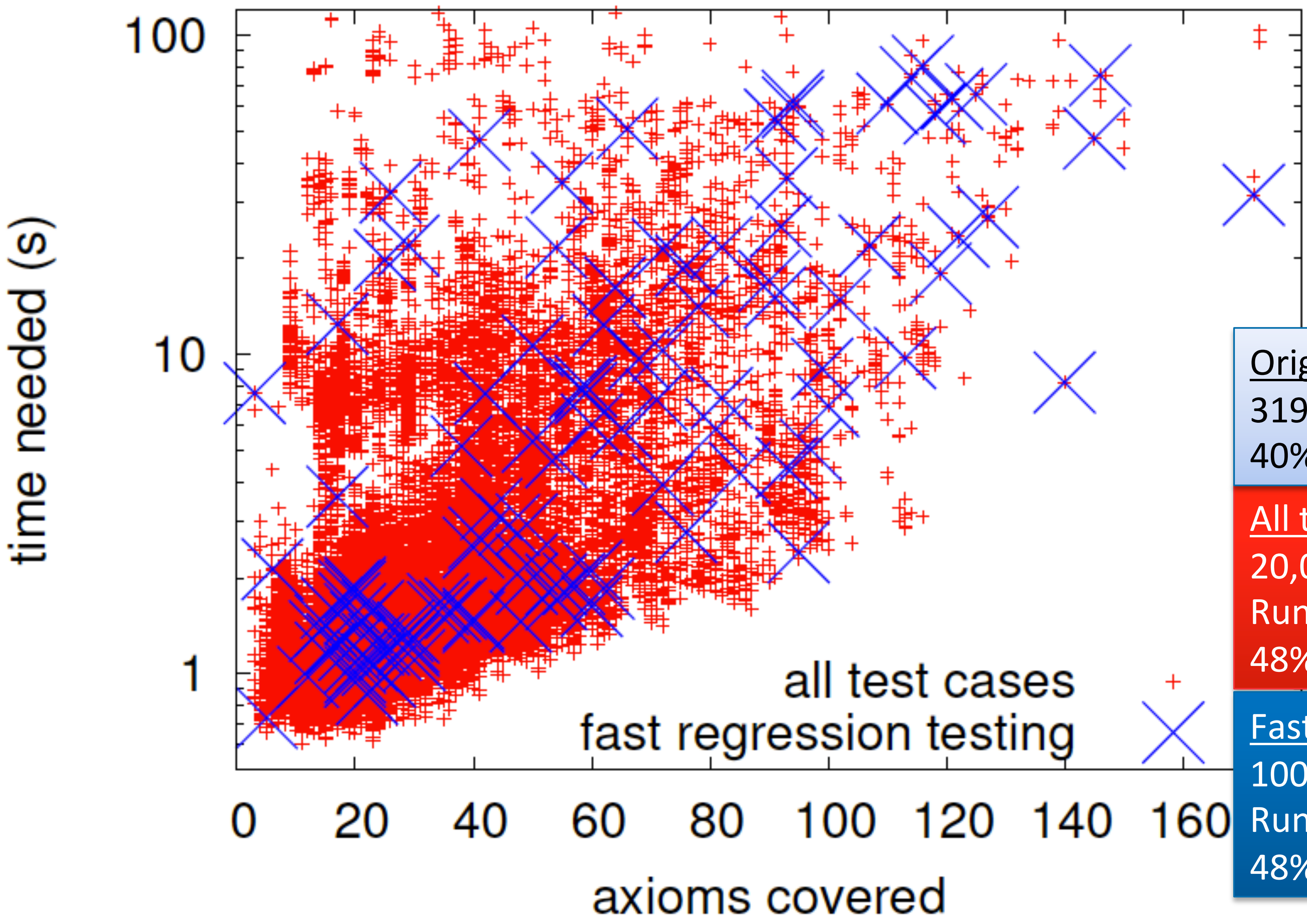
- Extremely fragile

Three simple heuristics to pick the "next axiom to drop":

[0.	Base case]	611 (40%)	
1.	Depth-first	701 (46%)	Naïve and good results
2.	Depth-first, ra	699 (46%)	More diverse sets
3.	Greedy (try to	688 (45%)	Often unsuccessful
4.	Breadth-first	687 (45%)	
5.	Breadth-first, random selection	684 (45%)	

→ Complimentary by design, verified by experiments.

Maximising Coverage & Minimising Time

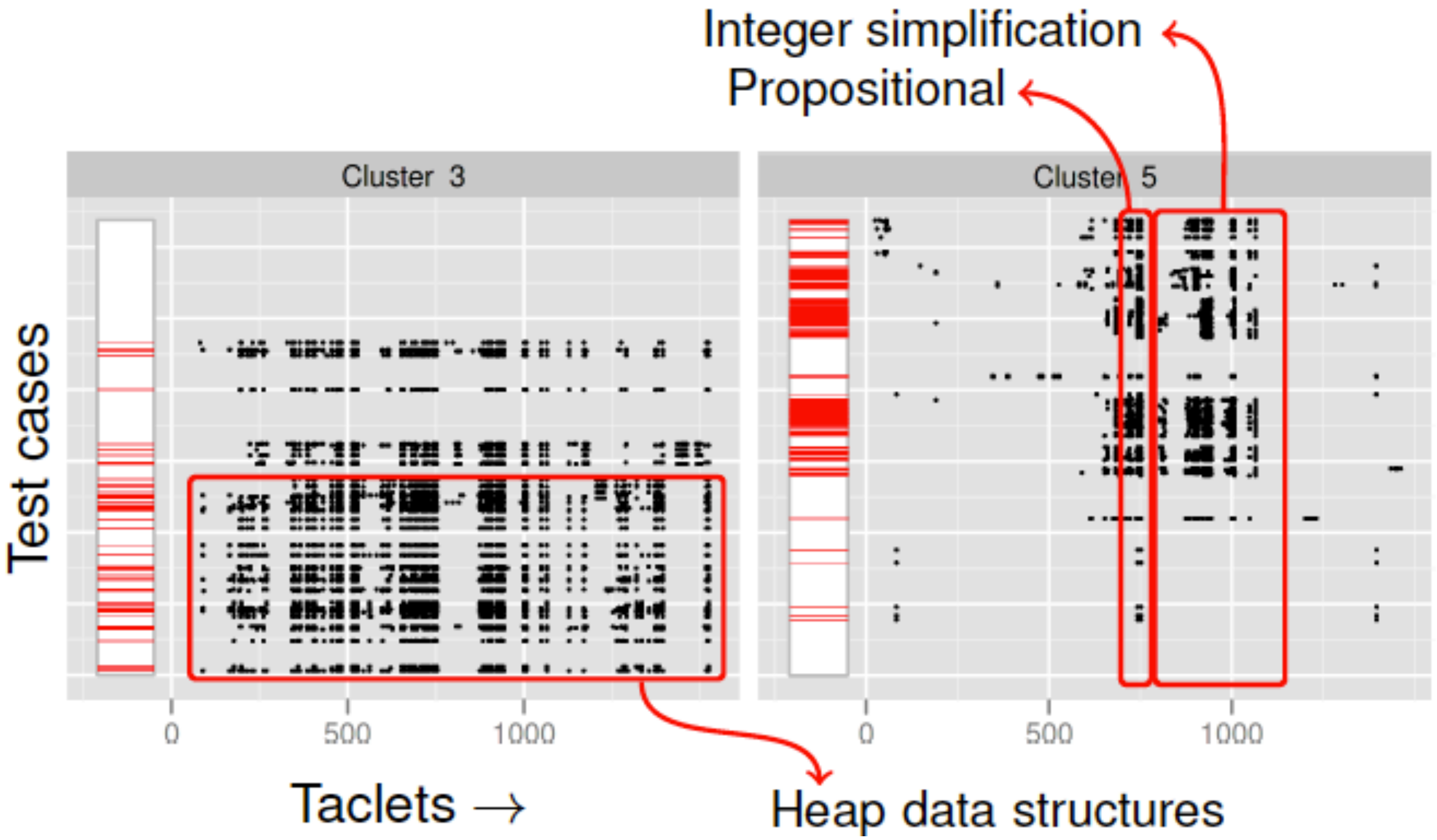


<u>Originally</u>
319 test cases
40% coverage

<u>All test cases</u>
20,000 test cases
Runtime: days
48% coverage

<u>Fast Regr. Testing</u>
100 test cases
Runtime: 1 h
48% coverage

What have we learned?



→ Problem understanding!

Take away

- We discovered that
  - Some KeY features are tested several times
  - Many KeY features are not tested (or are they unnecessary?)

- We hope
  - to discover bugs in the axiomatisation
  - to achieve 100% coverage (specialised test cases needed)

→ Comprehensive testing is necessary to achieve certain certifications.