

Applying Automata Learning to Embedded Control Software

Wouter Smeenk, **Joshua Moerman**,
Frits Vaandrager and David N. Jansen



A CANON COMPANY

Radboud University



Contributions

Completely learn industrial software

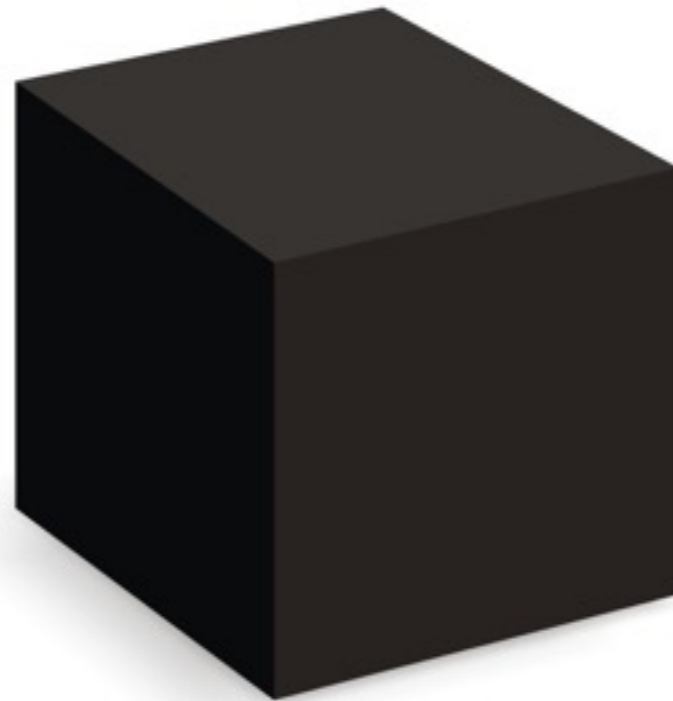
Novel FSM-based conformance testing method

Why models?

- Testing (new) implementations,
i.e. **conformance testing**
- Testing properties of the system,
i.e. **model checking**
- Derive implementations automatically,
i.e. **code generation**
- ...

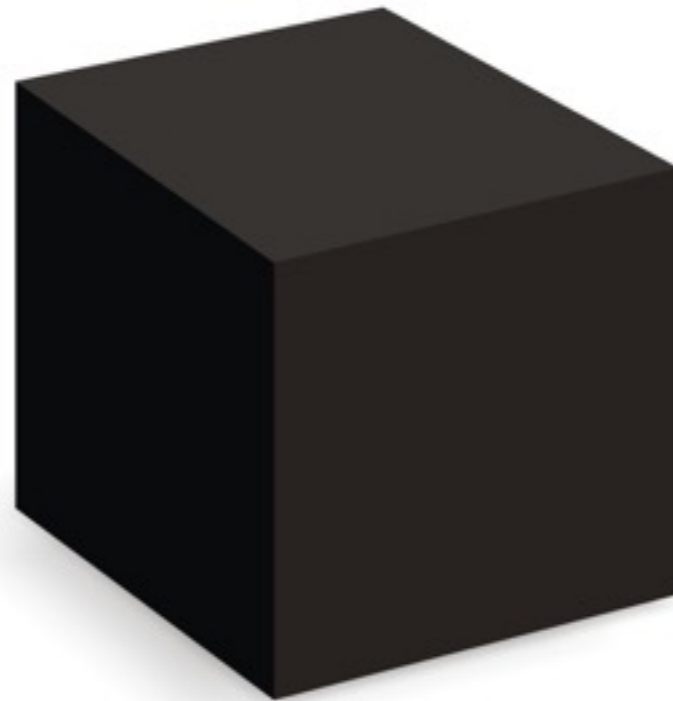
Reality

We don't have models, only black boxes...



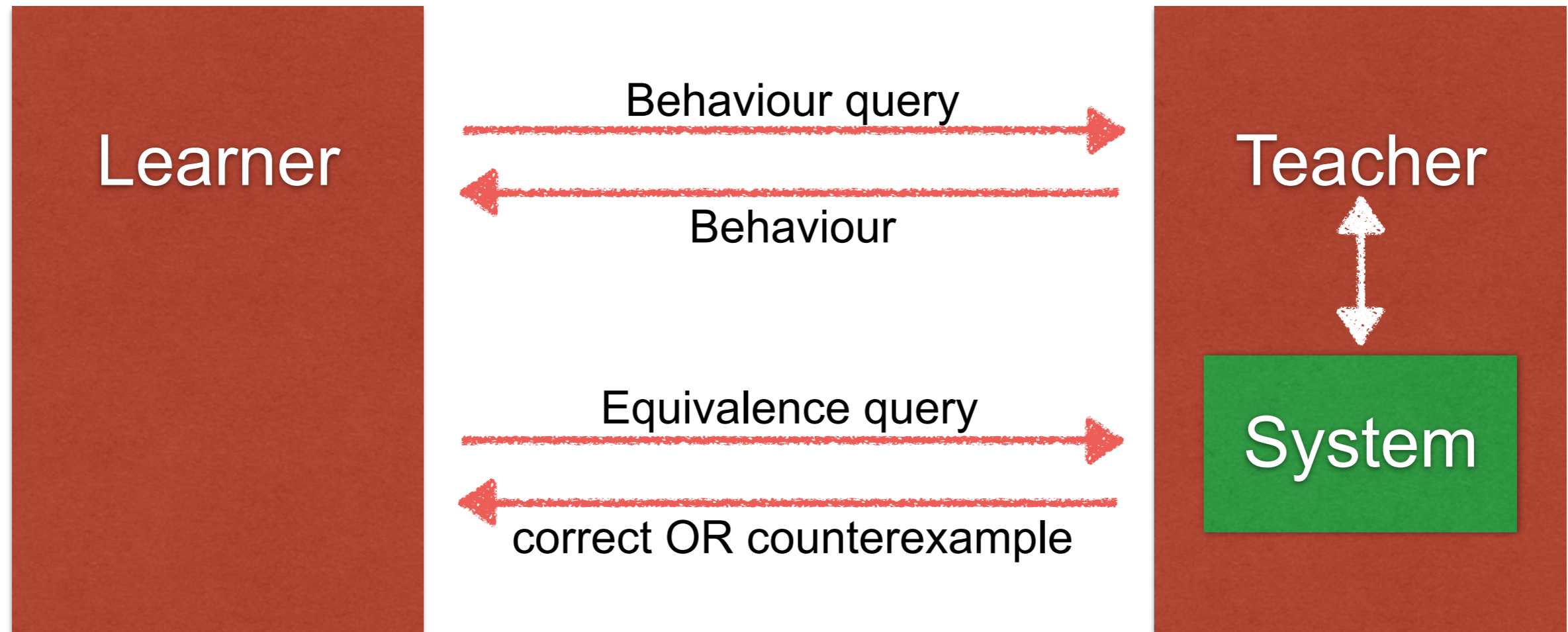
Reality

We don't have models, only black boxes...

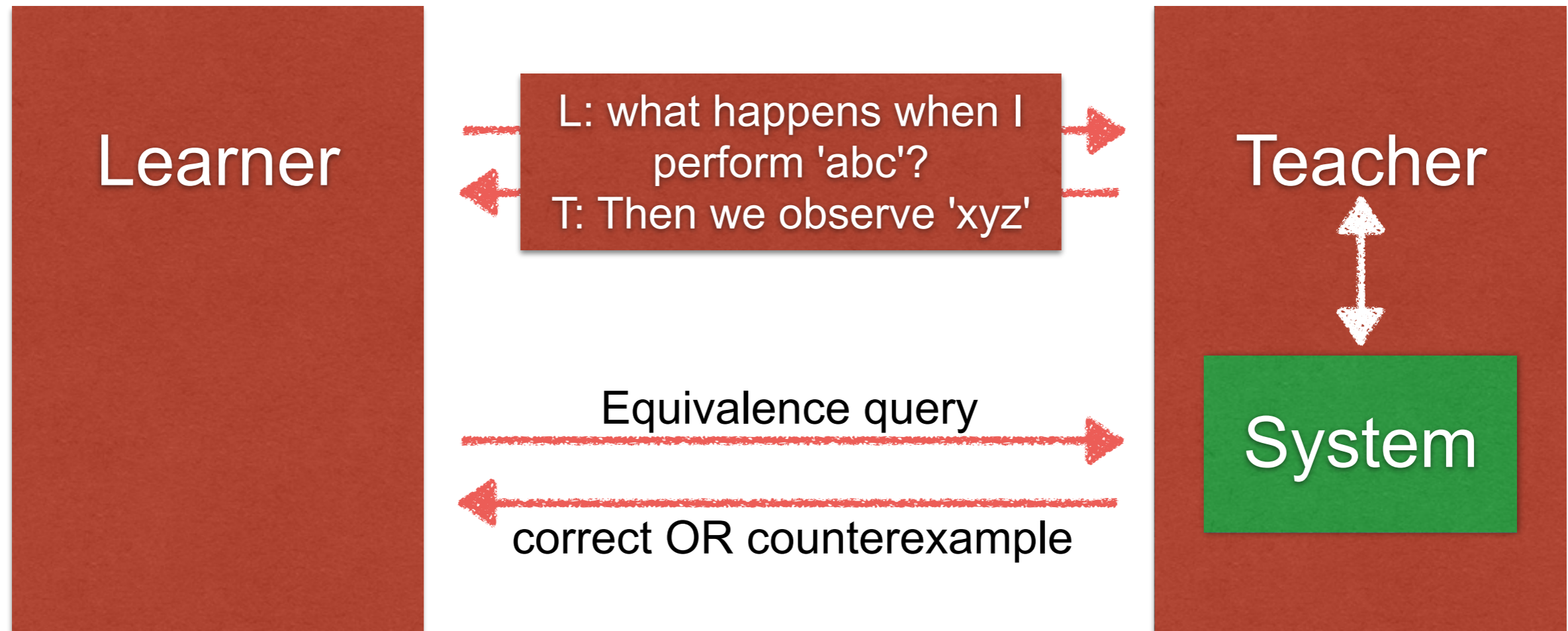


Goal: Automatically learn a model

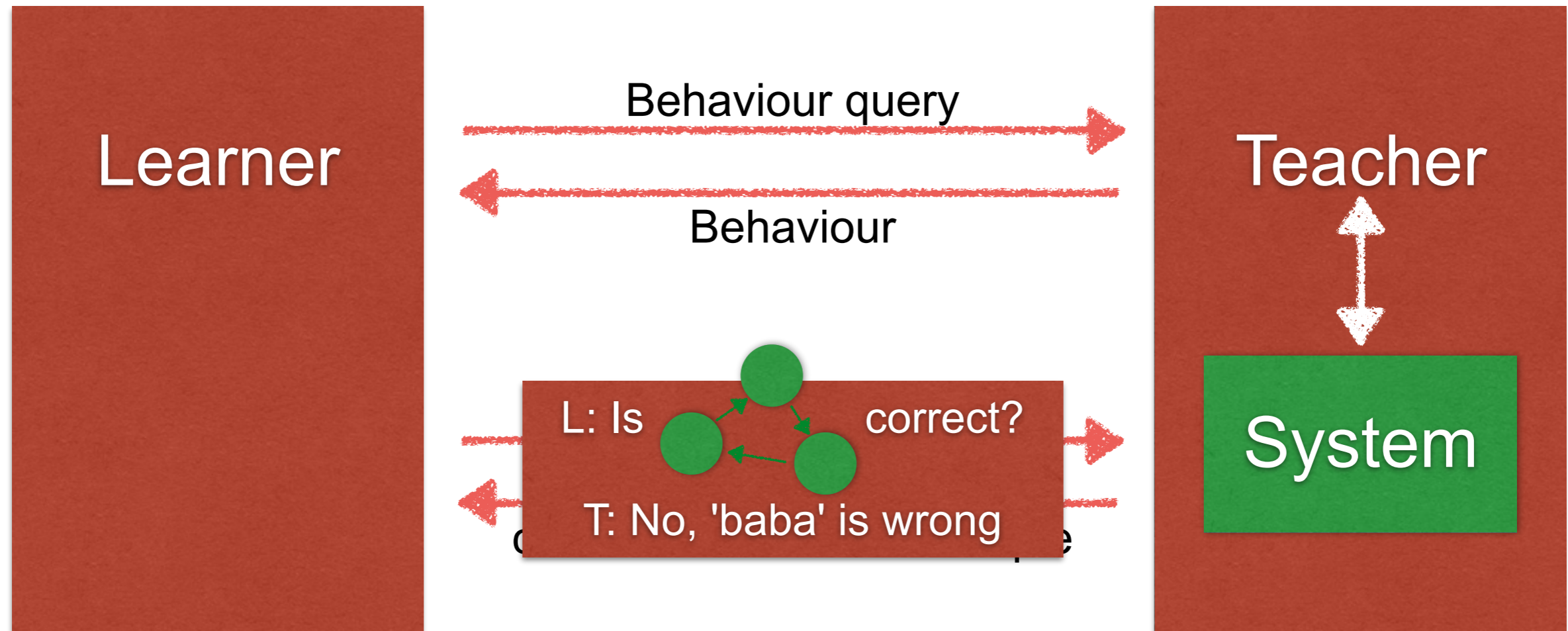
Active learning in theory



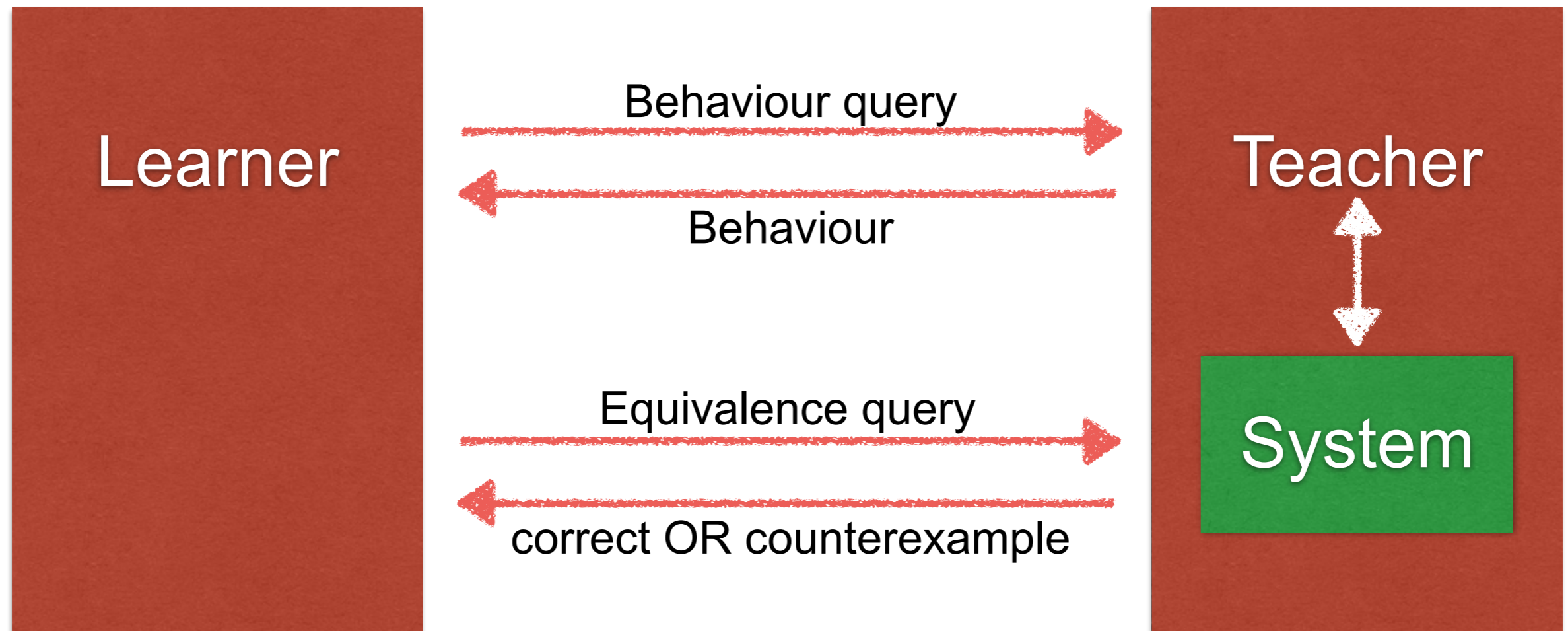
Active learning in theory



Active learning in theory

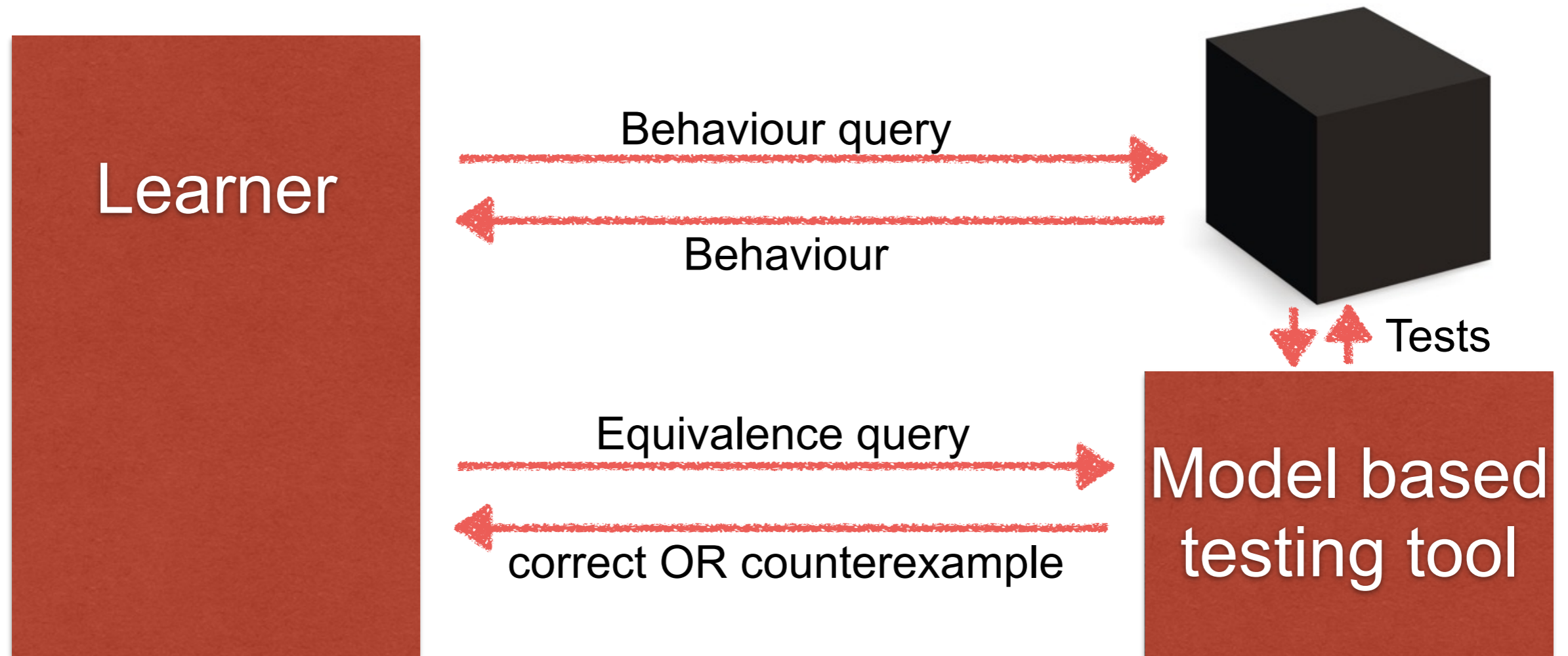


Active learning in theory

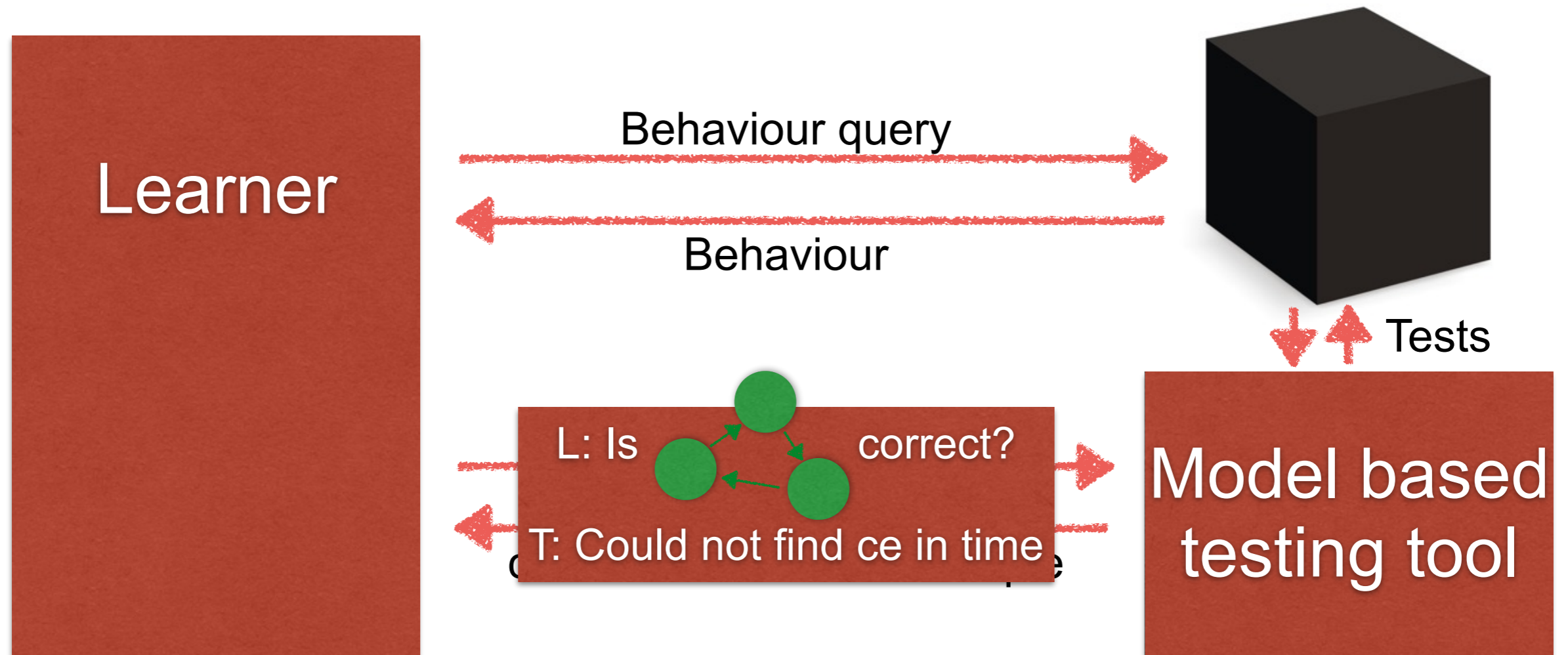


But: Who is the teacher?

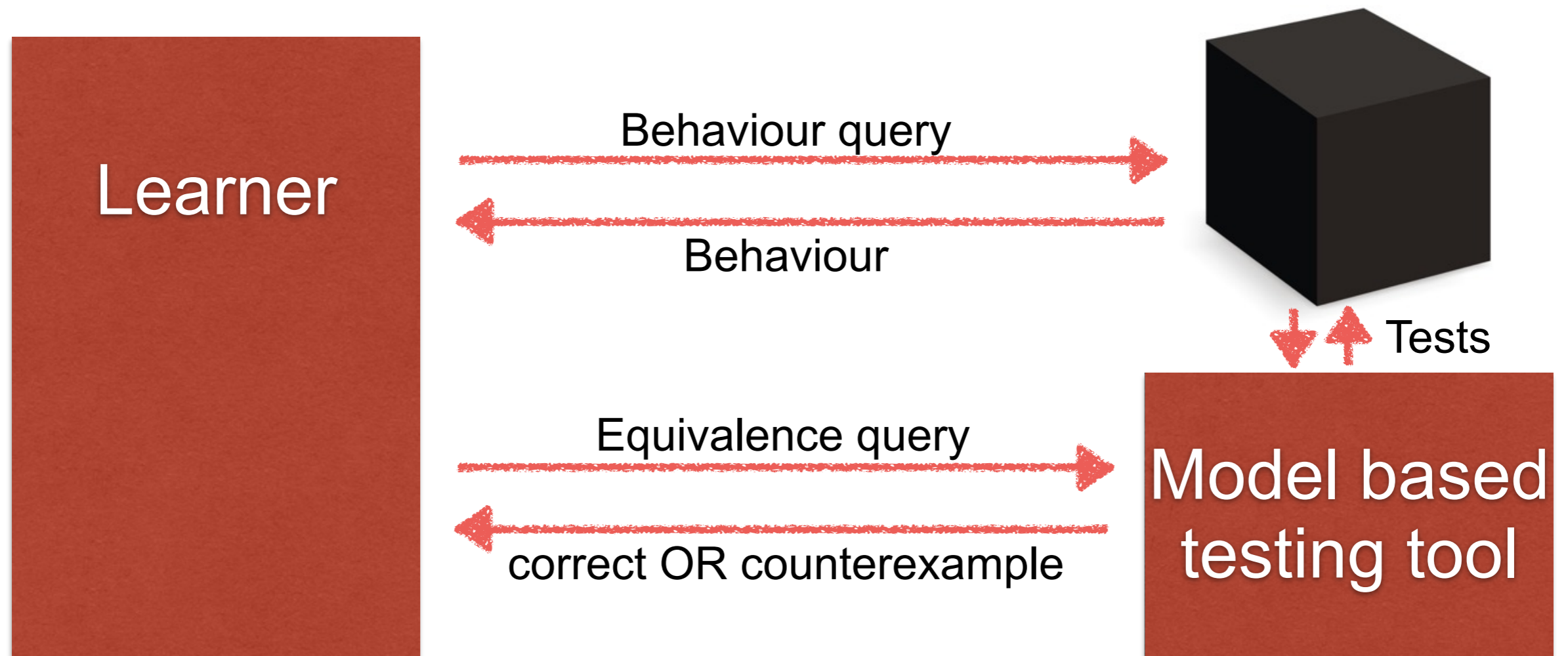
Active learning in practice



Active learning in practice



Active learning in practice



Learner is implemented by LearnLib

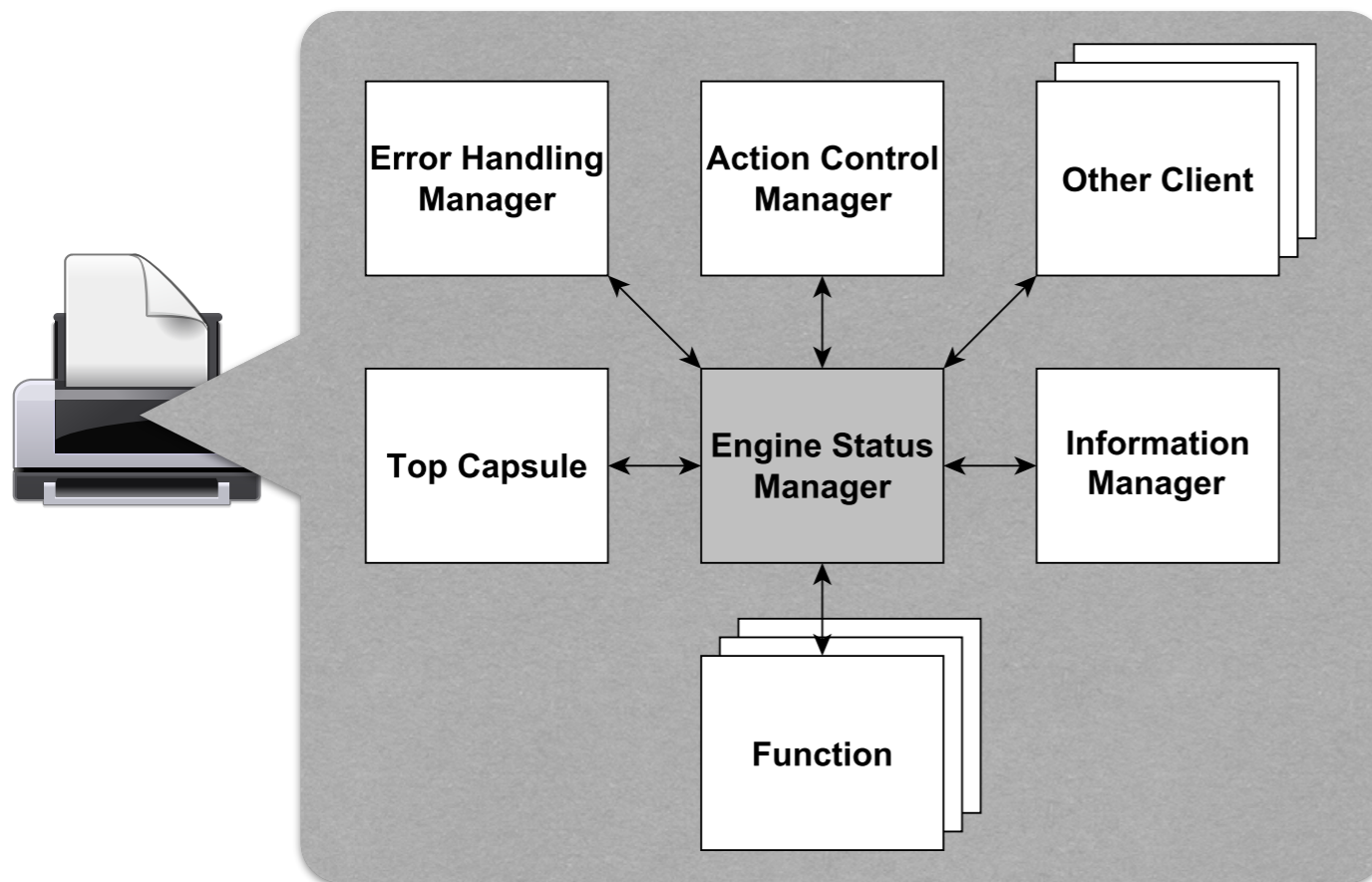
Our system

Printer software from Océ



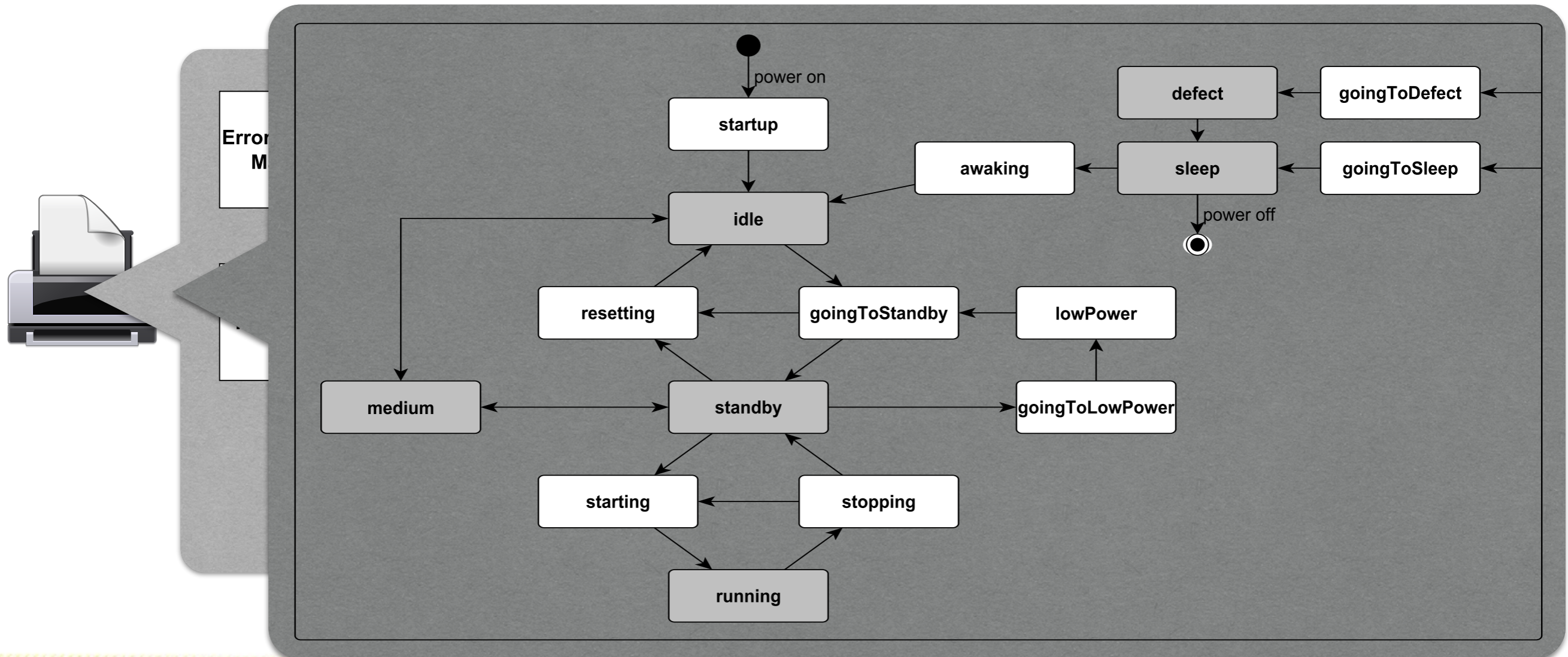
Our system

Printer software from Océ



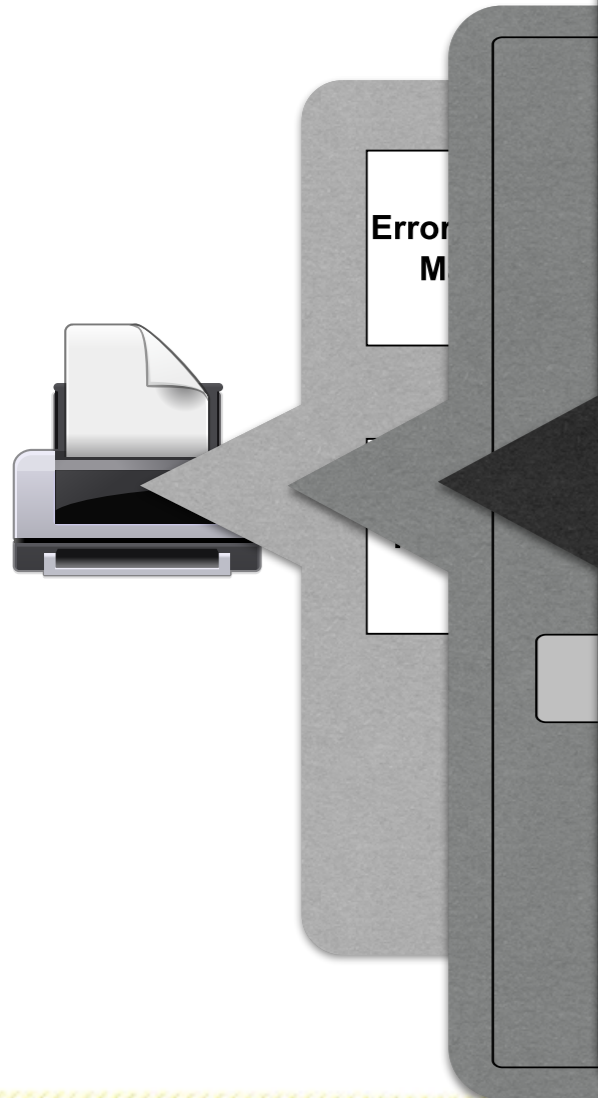
Our system

Printer software from Océ

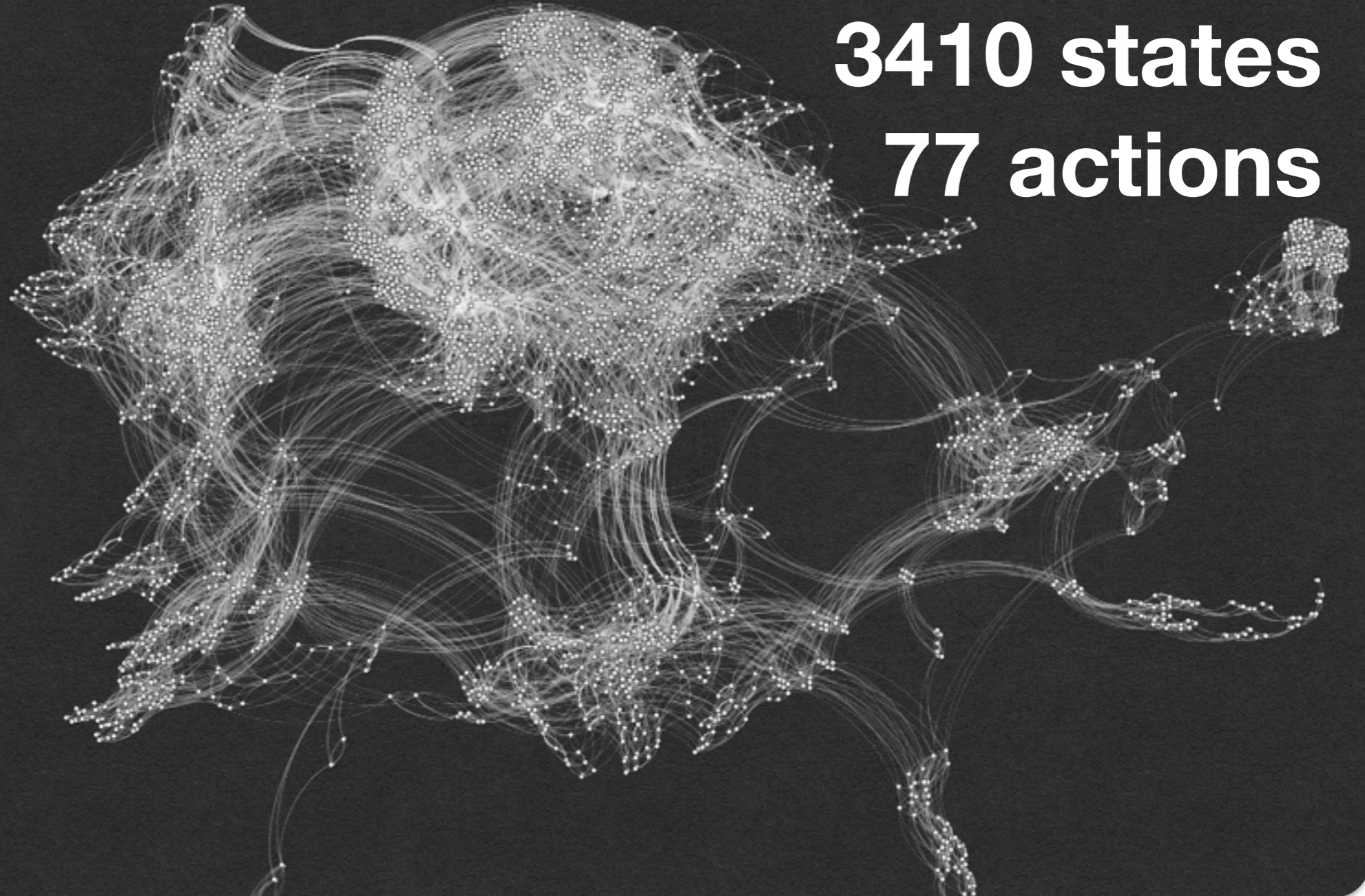


Our system

Printer so

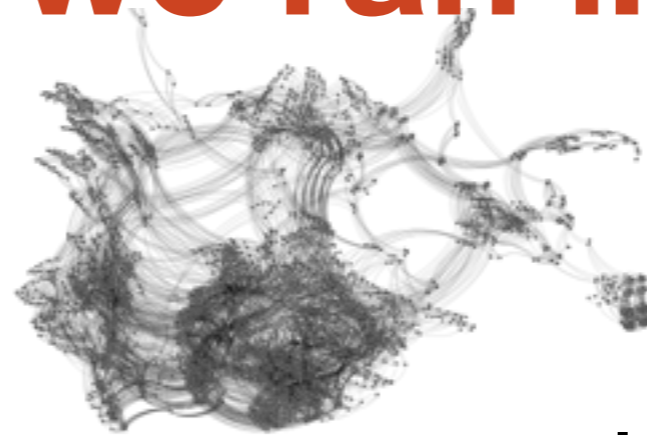


3410 states
77 actions



The problem we ran into

Given a hypothesis:



How do we find a counter example as fast as possible?

The problem we ran into

Given a hypothesis:



How do we find a counter example as fast as possible?

- Random walk?

The problem we ran into

Given a hypothesis:



How do we find a counter example as fast as possible?

- Random walk?
- W-method?

The W-method (and variants)

The W-method (and variants)

Prefix: bring the system to a specific state
(shortest path in hypothesis)

The W-method (and variants)

Prefix: bring the system to a specific state
(shortest path in hypothesis)

Suffix: test specific behaviour of a state
(derived from hypothesis)

The W-method (and variants)

Prefix: bring the system to a specific state
(shortest path in hypothesis)

Middle: to discover extra states
(exhaustive search)

Suffix: test specific behaviour of a state
(derived from hypothesis)

The W-method (and variants)

Prefix: bring the system to a specific state
(shortest path in hypothesis)

Middle: to discover extra states
(exhaustive search)

Suffix: test specific behaviour of a state
(derived from hypothesis)

$$P \cdot A^{\leq k} \cdot W$$

Two problems with $P \cdot A^{\leq k} \cdot W$

Recall that $|A| = 77$, so $A^{\leq k}$ will be big.

Two problems with $P \cdot A^{\leq k} \cdot W$

Recall that $|A| = 77$, so $A^{\leq k}$ will be big.

- Instead of $A^{\leq k}$, randomly sample A^* .

Two problems with $P \cdot A^{\leq k} \cdot W$

Recall that $|A| = 77$, so $A^{\leq k}$ will be big.

- Instead of $A^{\leq k}$, randomly sample A^* .

The set W is big with a lot of short words

Two problems with $P \cdot A^{\leq k} \cdot W$

Recall that $|A| = 77$, so $A^{\leq k}$ will be big.

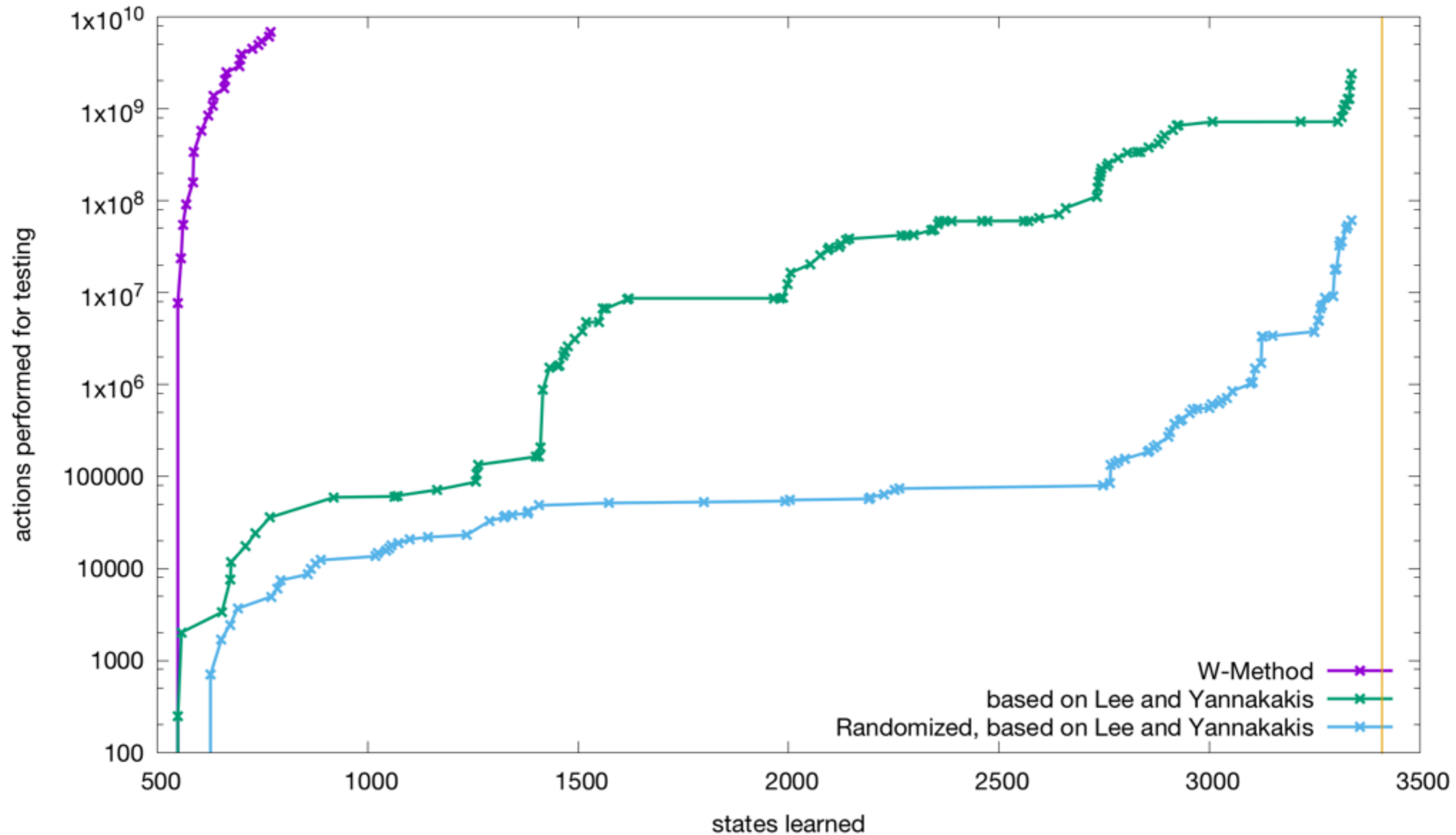
- Instead of $A^{\leq k}$, randomly sample A^* .

The set W is big with a lot of short words

- We adopted an algorithm by Lee and Yannakakis. This generates less, but longer suffixes.

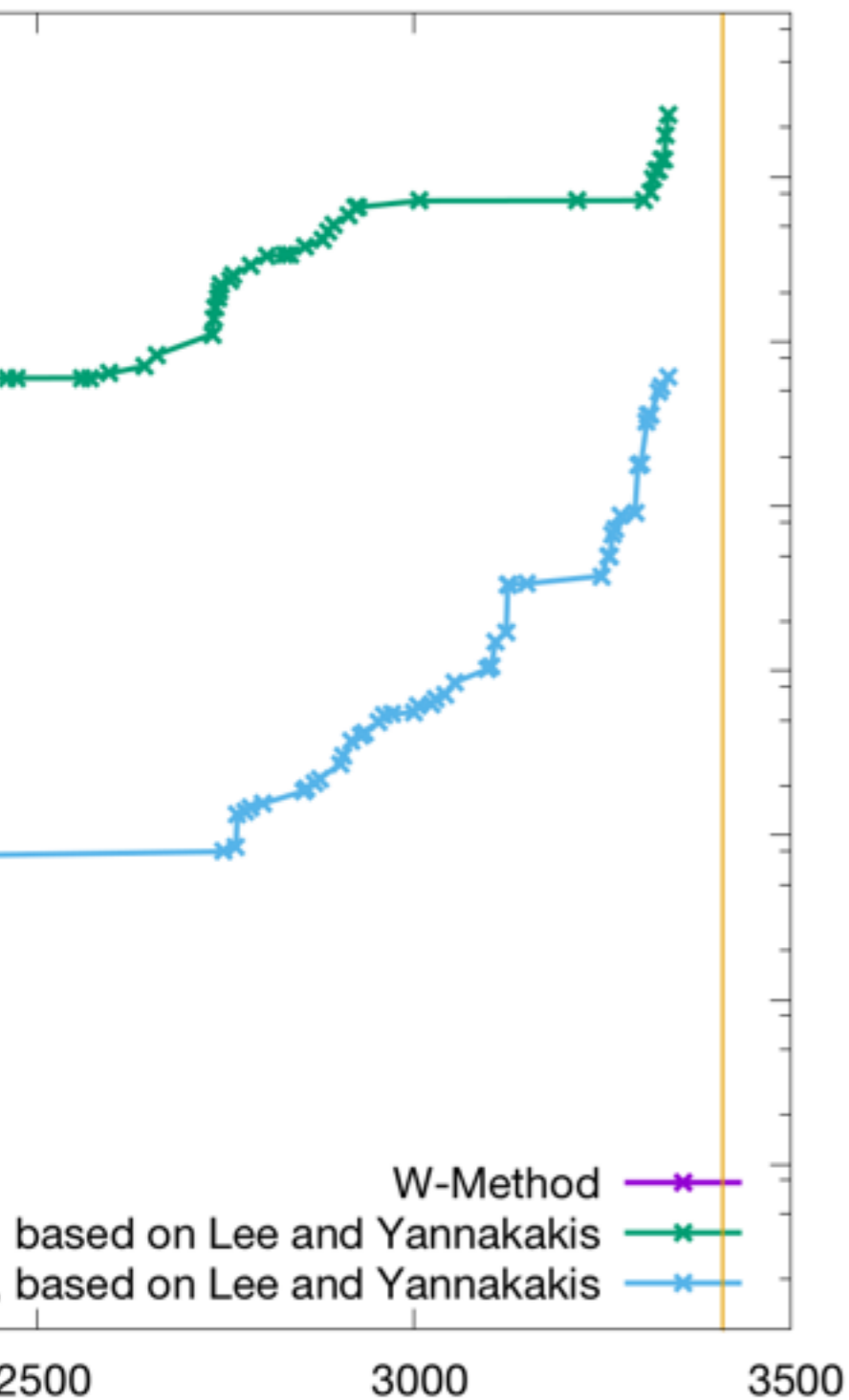
Results

One run of the learning process with three different test methods



Results

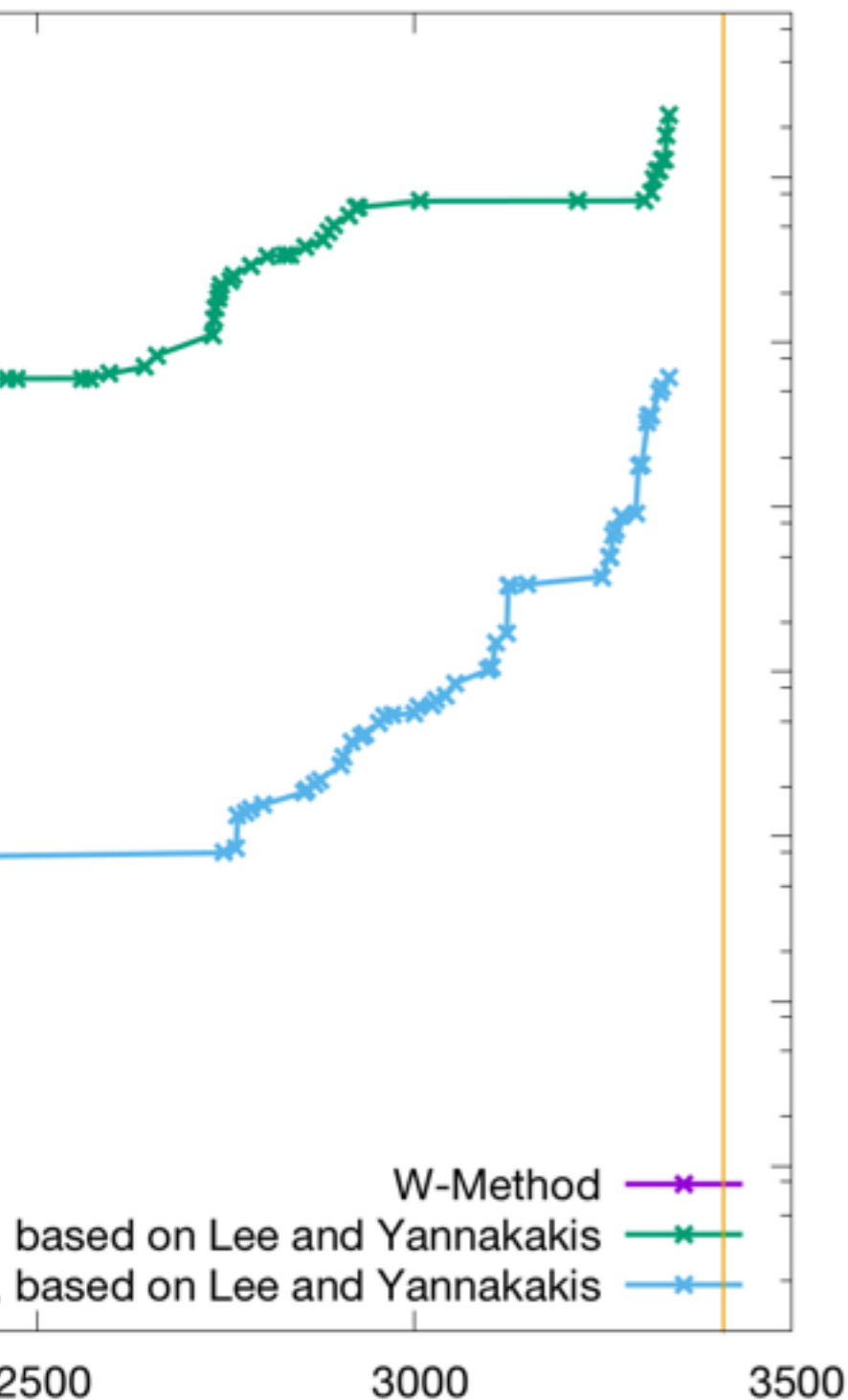
at test methods



Still not able to learn!

Results

at test methods

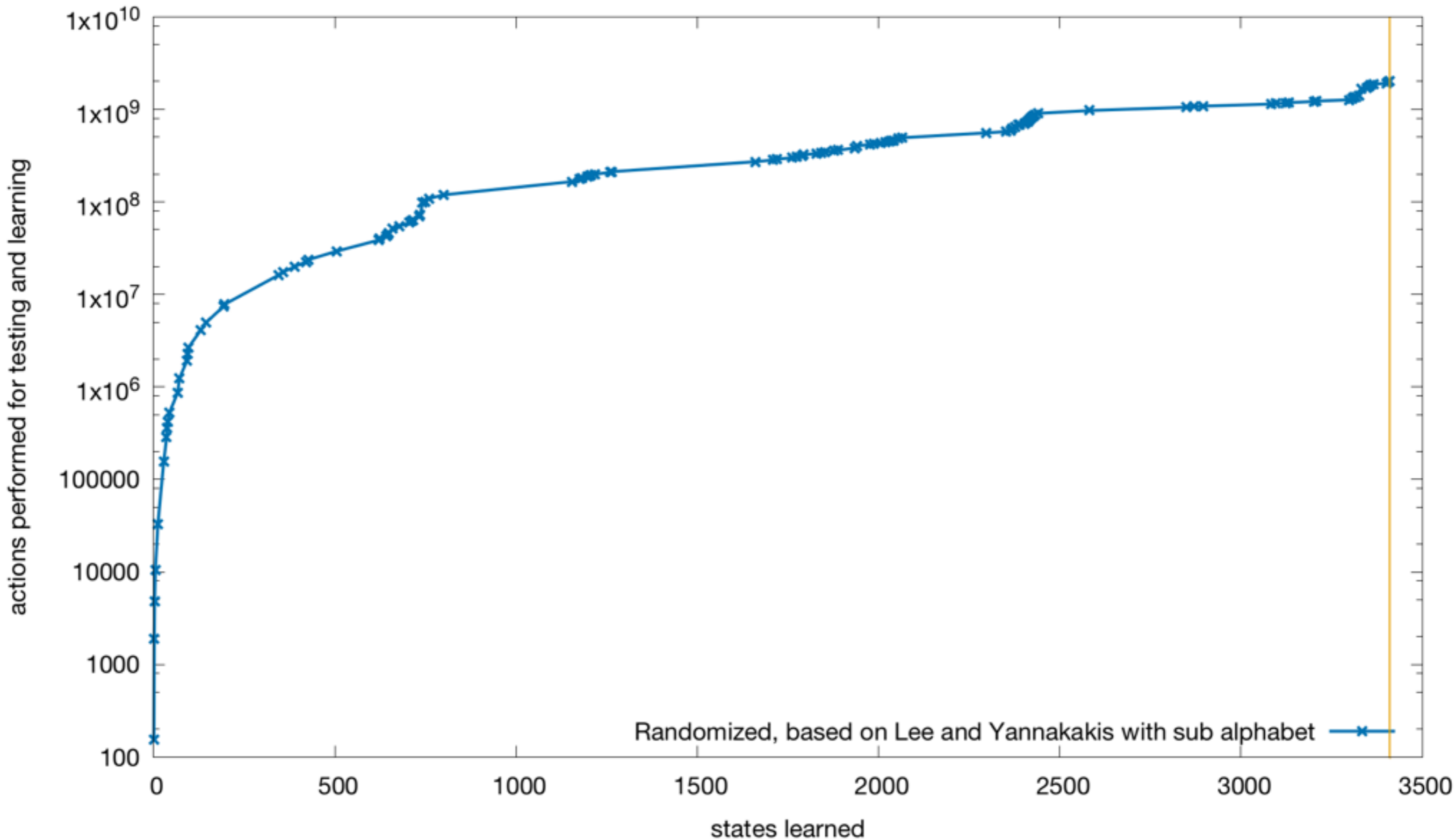


Still not able to learn!

Solution: Manually select sub alphabet for thorough testing

Final results

Run of the learning process with final setup on actual software



Conclusions

- We indeed can learn big black box systems

Conclusions

- We indeed can learn big black box systems
- With no (significant) use of expert knowledge

Conclusions

- We indeed can learn big black box systems
 - With no (significant) use of expert knowledge
-
- Applying Automata Learning to Embedded Control Software
Smeenk, W., Moerman, J., Vaandrager, F., Jansen, D.N., ICFEM 2015

Conclusions

- We indeed can learn big black box systems
- With no (significant) use of expert knowledge

- Applying Automata Learning to Embedded Control Software
Smeenk, W., Moerman, J., Vaandrager, F., Jansen, D.N., ICFEM 2015
- Lessons learned in the analysis of the EMV and TLS security protocols
J. de Ruijter, PhD thesis, 2015

Future work

- Still need for smarter test methods

Future work

- Still need for smarter test methods
- Learning of higher-level models

Future work

- Still need for smarter test methods
- Learning of higher-level models
- Usage of system logs during learning

Future work

- Still need for smarter test methods
- Learning of higher-level models
- Usage of system logs during learning
- Test our novel test method on its own right

Future work

- Still need for smarter test methods
- Learning of higher-level models
- Usage of system logs during learning
- Test our novel test method on its own right
- Integrate learning, testing and model checking

See talk on learning
TCP for abstractions
and model checking!

Future work

- Still need for smarter test methods
- Learning of higher-level models
- Usage of system logs during learning
- Test our novel test method on its own right
- Integrate learning, testing and model checking

Thank you!