An active learning algorithm automatically infers a behavioral model of a system by asking queries.

In each iteration of the learning algorithm, the learner constructs a hypothesis model of the system.

A system under learning, or SUL, is a reactive system for which we can apply inputs and observe outputs.

A model based tester (MBT) checks if the hypothesis is correct by checking it against the system.

A state machine, or automaton, is a transition model that contains states, which are connected by labeled transitions.

Abstraction is key when learning models of real-world systems. Manual abstraction is time-consuming. With Tomte we can do abstractions automatically.

Very large state machine for industrial controller

Standards describing network protocols typically fail to specify what an agent should do in case another agent does not follow the rules of the protocol. As a consequence, implementations of these standards may differ, which can result in security vulnerabilities.

We have shown that different implementations of TCP in Windows and Ubuntu induce different state machine models. Inspection of the learned models reveals that both implementations violate RFC 793.

In systems engineering, a potential bug in the far-away future is less troubling than a potential bug today.

For industrial control systems, the state space can be very large. So large that current state-of-the-art techniques and tools might not be powerful enough to learn complete, correct models.

In recent work, however, we present an algorithm that ensures that for subsequent hypotheses the minimal length of a counterexample never increases, which implies that the distance to the target never increases in a corresponding ultrametric. This way we get the best model we have seen so far as a hypothesis.


Active Learning of State Machines: Theory and Practice
Paul Fiterau-Brostean and Rick Smetsers

NETWORK PROTOCOLS

For more information on our research group, visit http://mbsd.cs.ru.nl.
For more information on the tools and techniques, see http://tomte.cs.ru.nl.

This work is partly supported by NWO and STW projects:
“Learning Extended State Machines for Malware Analysis” (LEMMA)
“Active Learning of Security Protocols” (ALSEP)
“Integrating Testing and Learning of Interface Automata” (ITALIA)

CONTROL SOFTWARE