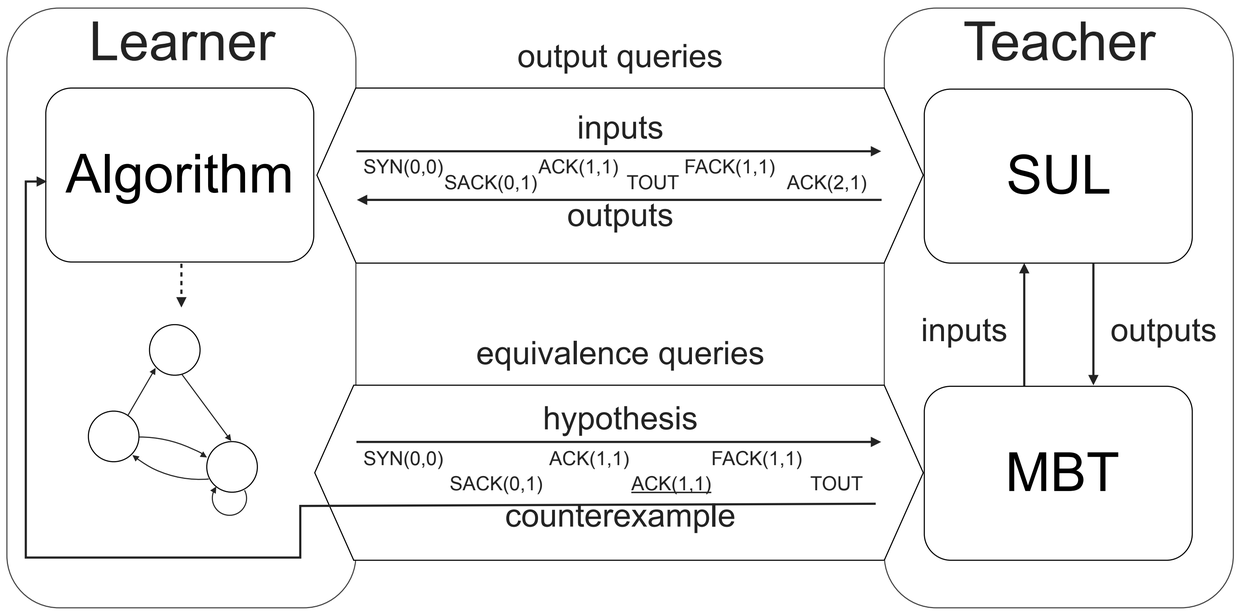
Active Learning of State Machines: Theory and Practice

Paul Fiterau-Brostean and Rick Smetsers

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

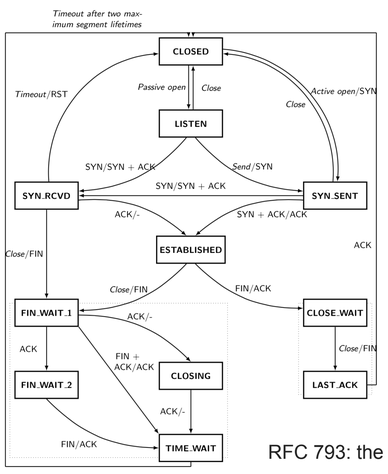
In each iteration of the learning algorithm, the learner constructs a **hypothesis** model of the system.



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.

LEARNING



RFC 793: the TCP standard.

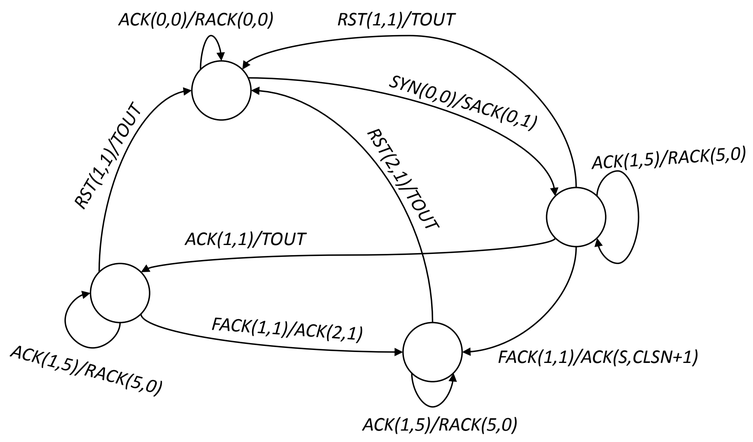
The expected behavior of the SUL is sometimes described in a **specification**. In this case we can compare this to the learned model.

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**.

Fiterău-Broştean, P., Jansen, R., & Vaandrager, F. (2014). Learning fragments of the TCP network protocol. In F. Lang & F. Flammini (Eds.), Proceedings of the 19th International Conference on Formal Methods for Industrial Critical Systems, FMICS '14 (Vol. 8718, pp. 78–93). Cham: Springer International Publishing.

NETWORK PROTOCOLS



State machine for TCP.

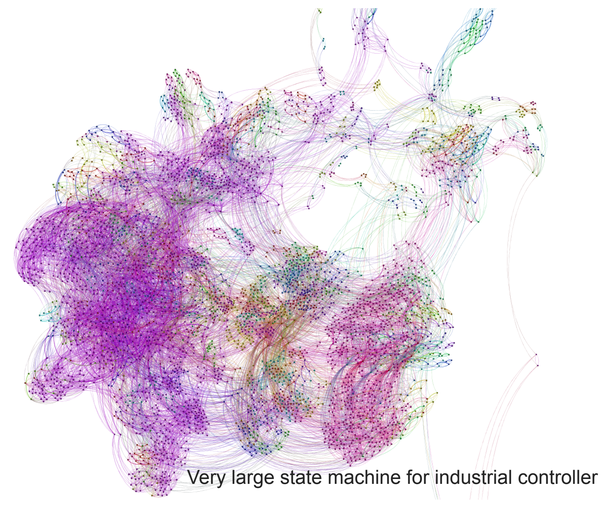
A **state machine**, or **automaton** is a transition model that contains **states**, which are connected by labeled **transitions**.

MODEL

Tomte bridging the learner and the SUL.

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

ABSTRACTION



Very large state machine for industrial controller

*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.

Smetsers, R., Volpato, M., Vaandrager, F., & Verwer, S. (2014). Bigger is not always better: on the quality of hypotheses in active automata learning. In Proceedings of the 12th International Conference on Grammatical Inference (pp. 167–181).

CONTROL SOFTWARE

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)

Radboud University