Learning algorithms are increasingly used in multiple disciplines of today’s computer science reaching from robotics and formal verification to bioinformatics or natural language recognition.

In this talk, we consider learning algorithms in the context of software development and verification. Two novel variants of learning algorithms will be presented. In the first part, we introduce NL* an algorithm for learning nondeterministic automata from queries and counterexamples. NL* will be shown to yield exponentially better results concerning number of states of the resulting automata than its deterministic version L*. Though theoretical worst case results concerning membership- and equivalence queries are worse than for L*, we will show that in practice NL* usually outperforms L* by far.

The second part of the presentation is concerned with learning and synthesizing distributed automata. To this end, we extend the well-known L* learning algorithm for deterministic automata. Many existing approaches learn a global communicating system and project it to its communicating processes. This, however, usually implies additional and most often undesired behavior. In contrast, we present an approach which yields exact systems, i.e., behavior that was specified as positive will always be contained in the final system and undesired behavior excluded.