

PRISMATIC summary

As a tool for verification, PRISMATIC provides several related functions. First, most generally, PRISMATIC is built on an enhanced version of the PRISM model checker, which supports probabilistic model checking of systems described as concurrent probabilistic state machines (www.prismmodelchecker.org). Second, PRISMATIC includes additional statistical model checking methods that can operate on those same system models, but may scale better for certain types of systems. Third, PRISMATIC includes culprit identification methods that can help diagnose which components in a model are responsible for *failing* to verify a particular property. And finally, PRISMATIC includes several tools that help to compose system models from separate component models, and support abstract component models that can be used in hierarchical designs, for scalability.

PRISM provides support for modeling and analysis of several different classes of probabilistic models: discrete and continuous-time Markov chains, Markov decision processes, and probabilistic timed automata. PRISM provides a flexible modeling language for describing those models, and supports a wide range of probabilistic temporal logics for specifying properties to be verified. Classic PRISM incorporates several efficient verification engines, primarily based on symbolic model checking techniques that use extensions of binary decision diagram (BDD) data structures.

The newer statistical verification methods included in PRISMATIC simply draw sample traces of the system's execution (e.g., from a simulation) and use statistical methods to test whether the accumulated samples justify a sufficiently-confident conclusion about whether the system's performance meets the requirements. If a conclusion is not yet possible, more samples are drawn. This approach is less sensitive to the state explosion problem that causes difficulties for exhaustive model checking algorithms.

In addition to analytic/numeric and statistical model checking methods in the enhanced PRISM tool itself, the PRISMATIC package includes several tools to support component-based modeling and verification. Stylized, partial PRISM models are used to represent the behavior of individual components. The XMC tool assembles component models into a full system model that PRISM can then reason about. The ABV tool assembles a specialized "abstraction verification" model that is used to prove that an abstract PRISM model is an admissible abstraction of a more detailed system model.

PRISMATIC also includes new client-server capabilities, so that the verification tool can be run as a parallel service in a cloud-computing environment such as Amazon's EC2.

PRISMATIC also has tools to provide *culprit identification*: determining, if a desired property has insufficiently large probability, which components of the design which are most likely to be responsible. To find culprits, PRISMATIC samples the paths through the model state space to obtain traces associated with both faulting and non-faulting simulations. It collects the values of the mode variables which are off-nominal in any sampled run, along with the boolean evaluation of the property for each sample, and passes them to a decision tree algorithm to identify components whose failures are most often associated with the specified system fault.