STeP: The Stanford Temporal Prover

Zohar Manna, Nikolaj Bjørner, Anca Browne, Edward Chang, Michael Colón, Luca de Alfaro, Harish Devarajan, Arjun Kapur, Jaejin Lee, Henny Sipma and Tomás Uribe

Computer Science Department, Stanford University
Stanford, CA 94305

The Stanford Temporal Prover, STeP, supports the computer-aided formal verification of reactive (and, in particular, concurrent) systems based on temporal specifications. Reactive systems maintain an ongoing interaction with their environment; their specifications are typically expressed as constraints on their behavior over time. Unlike most systems for temporal verification, STeP is not restricted to finite-state systems, but combines model checking with deductive methods to allow the verification of a broad class of systems, including parameterized (N-component) circuit designs, parameterized (N-process) programs, and programs with infinite data domains. In short, STeP has been designed with the objective of combining the expressiveness of deductive methods with the simplicity of model checking.

STeP verifies temporal properties of systems by means of verification rules and verification diagrams. *Verification rules* are used to reduce temporal properties of systems to first-order verification conditions [MP95]. *Verification diagrams* [MP94] provide a visual language for guiding, organizing, and displaying proofs. Verification diagrams allow the user to construct proofs hierarchically, starting from a high-level, intuitive proof sketch and proceeding incrementally, as necessary, through layers of greater detail.

The system implements powerful techniques for automatic *invariant generation*. Deductive verification almost always relies on finding, for a given program and specification, suitably strong (inductive) invariants and intermediate assertions. The user can typically provide an intuitive, high-level invariant, from which the system derives stronger, more detailed, *top-down invariants*. Simultaneously, *bottom-up invariants* are generated automatically by analyzing the program text. By combining these two methods, the system can often deduce sufficiently detailed invariants to carry through the entire verification process.

The system also provides an integrated suite of simplification and decision procedures for automatically checking the validity of a large class of first-order and temporal formulas. This degree of automated deduction is sufficient to handle most of the verification conditions that arise in deductive verification.

An overview of STeP is shown in Figure 1. The main inputs are a reactive system (which can be a hardware or software description) and a property to be proven about the system, represented by a temporal logic formula. Verification can be performed either by the model checker or by deductive means. In the latter case, the proof is typically automatic for safety properties. The proof of progress properties may require user guidance, provided by verification diagrams.

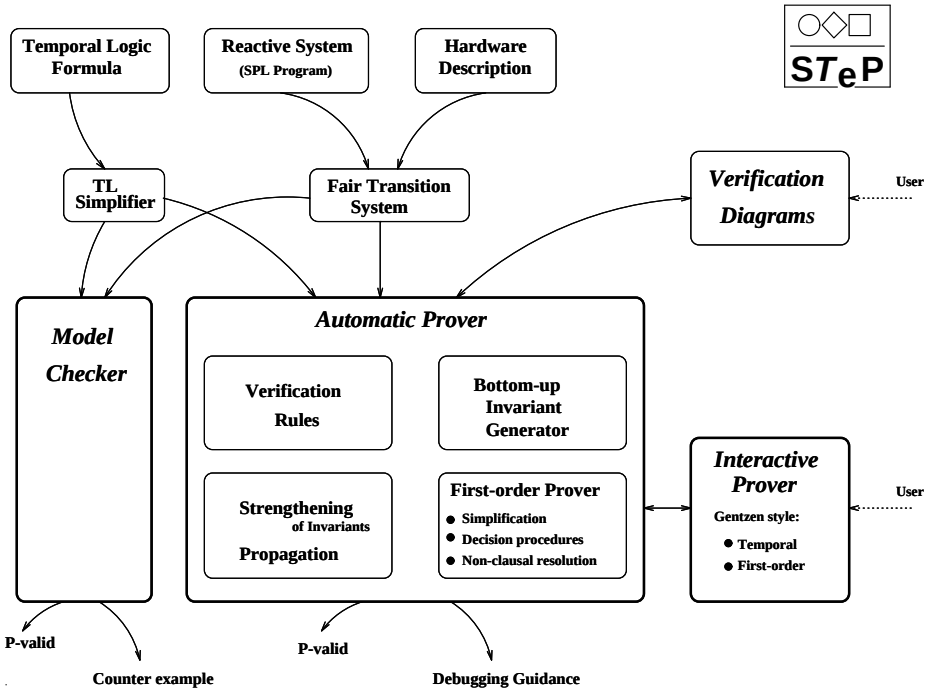

Fig. 1. An overview of the STeP system

In either case, the automatic prover is responsible for generating and proving the required verification conditions. An interactive Gentzen-style theorem prover and a resolution-based prover are available to establish the few verification conditions that are not proved automatically. For a more extensive description of the STeP system and examples of verified programs, the reader is referred to [MAB+94].

STeP is implemented in Standard ML of New Jersey, using CML and eXene for its X-windows user interface. An educational version of the system is currently available. For information on obtaining the system, send e-mail to `step-request@cs.stanford.edu`.

References

[MAB+94] Z. Manna, A. Anuchitanukul, N. Bjørner, A. Browne, E. Chang, M. Colón, L. de Alfaro, H. Devarajan, H. Sipma, and T. Uribe. STeP: the Stanford Temporal Prover. Technical report STAN-CS-TR-94-1518, Computer Science Department, Stanford University, June 1994.

[MP94] Z. Manna and A. Pnueli. Temporal verification diagrams. In *Proc. of the Int. Symp. on Theoretical Aspects of Computer Software*, volume 789 of *LNCS*, pages 726–765. Springer-Verlag, April 1994.

[MP95] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety.* Springer-Verlag, New York, 1995.