Beyond Model Checking

Zohar Manna

Department of Computer Science
Stanford University
Stanford, CA 94305

This talk will describe STEP (the Stanford TEmporal Prover), a system to support computer-aided verification of reactive systems based on their temporal specifications. Unlike most systems for temporal verification, STEP does not concentrate solely on finite-state systems. It combines model checking with algorithmic deductive methods (decision procedures) and interactive deductive methods (theorem proving). The user is expected to interact with the system and provide, whenever necessary, top-level guidance in the form of auxiliary invariants for safety properties, and well-founded measures and intermediate assertions for progress properties.

A verification system which combines model checking and deductive methods can offer a number of advantages over purely model checking or purely deductive approaches. Such a system should

- Avoid the state-explosion problem by decomposing the verification of large systems into the verification of smaller system components, where each component may be verified by a different method.
- Use deduction rules, if appropriate, to replace the global verification of a system component by local proofs of verification conditions.
- Free the user from having to deal with low-level details of verification, which are handled through model checking or decision procedures, and
- Allow the verification of infinite-state systems, such as parameterized systems or systems with infinite data domains.

In short, STEP has been designed with the objective:

> To combine the expressiveness of deductive methods with the simplicity of model checking.

Development efforts have been focused, in particular, in the following areas.

First, in addition to the textual language of temporal logic, the system supports a structured visual language of *verification diagrams*[1] for guiding 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.

Second, the system implements powerful techniques (algorithmic and heuristic) for automatic *invariant generation*. Deductive verification in the temporal framework almost always relies heavily on finding, for a given program and specification, suitably

[1] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems.* Springer-Verlag, New York, 1994. To appear.

strong invariants and intermediate assertions. The user can typically provide an intuitive, high-level invariant, from which the system derives stronger, *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.

Finally, the system provides a built-in facility for automatically checking a large class of first-order and temporal formulas, based on simplification methods, term rewriting, and decision procedures. This degree of automated deduction is sufficient to handle most of the verification conditions that arise during the course of deductive verification — and the few conditions that are not solved automatically correspond to the critical steps of manually constructed proofs, where the user is most capable of providing guidance.

Many of the examples in the Manna-Pnueli textbook *Temporal Verification of Reactive Systems*[2] have been automatically verified using STEP. These include resource allocation protocols based on message-passing and on shared variables, a solution to the readers-writers problem, examples using infinite data domains, and a parameterized solution for the N-process dining philosophers problem. We have also automatically verified a substantially more complex parameterized program, Szymanski's N-process mutual exclusion protocol.

[2]Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems.* Springer-Verlag, New York, 1994. To appear.