Formal Aspects of Computing (2000) 12: 218–219

Formal Aspects of Computing

The 'Cash-Point' Service: A Verification Case Study Using STeP

Anca Browne, Bernd Finkbeiner, Zohar Manna and Henny Sipma

Computer Science Department, Stanford University, Stanford, CA, USA

Abstract. STeP, the Stanford Temporal Prover, supports the computer-aided formal verification of concurrent and reactive systems based on temporal specifications [MBB99]. Automated *model checking* is combined with computer-aided *deductive methods* to allow for the verification of a broad class of systems, including parameterised (N-component) circuit designs, parameterised (N-process) programs, and programs with infinite data domains.

Keywords: Model checking; STeP; Verification; Verification diagrams

1. System Description

STeP provides a modular language for hierarchical system descriptions [FMS98]. In the case study, the cash-point service consists of a number of *tills* that can access a central *database* through *network* connections:

$$\left(||_{i=1..N}(\texttt{till} \parallel \texttt{network})\right) \parallel \texttt{database}$$

The system is parameterised by the unbounded number of tills N. The top-level components are again constructed from smaller building blocks: the network, for example, consists of two data lines (one from the till to the database and one in the opposite direction). The data lines in turn are constructed from buffers and protocol components, e.g. for authentication. Parts of the system can be modelled as finite-state components (e.g. the network protocols), while others have an infinite state-space. This includes discrete components like the database as well as real-time components like the tills.

2. Property Specification

STeP's specification language is linear time temporal logic. Properties are locally assigned to the modules that guarantee their validity. In the case study, the properties cover a variety of different types, including safety (such as correct authentication in the network), liveness (e.g. responsiveness of the database) and real-time properties (e.g. response time properties of the till).

Correspondence and offprint requests to: Zohar Manna, Computer Science Department, Stanford, Stanford University, CA 94305, USA. Email: finkbein@theory.stanford.edu

3. Component Analysis

Since the protocol components of the cash-point network have a finite state-space, they are automatically verified using *model checking*. Other modules, with a very large finite or infinite state-space, are analysed using STeP's deductive methods: *Verification rules* [MP95] reduce simple temporal properties to first-order verification conditions. *Verification diagrams* [MP94, BMS95, MBS98] reduce arbitrary temporal properties to first-order verification conditions and a finite-state temporal verification condition that can be model checked. STeP provides a collection of simplification and decision procedures that automatically check the validity of a large class of first-order and temporal formulas [Bjø98].

4. Compositional Analysis

Many global system properties can be decomposed into guarantees of the individual components that, together, imply the property. The overall responsiveness of the cash-point service, for example, relies on the responsiveness properties of the till, the network and the database. After the local properties have been shown to hold with the most appropriate method for the given component, they are automatically *inherited* as lemmas for the global system. Verification diagrams are an elegant method to combine these results into a proof for the global property.

5. Conclusions

The parameterisation and the variety of component and property types make the cash-point service a challenging case study for a verification tool like STeP. The case study demonstrates the use of different verification methods for different component types and their combination into global results using property inheritance and verification diagrams.

Acknowledgements

This research was supported in part by the National Science Foundation under grant CCR-98-04100 and CCR-99-00984 ARO under grants DAAH04-96-1-0122 and DAAG55-98-1-0471, ARO under MURI grant DAAH04-96-1-0341, by Army contract DABT63-96-C-0096 (DARPA), and by Air Force contract F33615-99-C-3014.

References

[Bjø98] Bjørner, N. S.: *Integrating Decision Procedures for Temporal Verification.* PhD thesis, Computer Science Department, Stanford University, November 1998.

[BMS95] Browne, A., Manna, Z. and Sipma, H. B.: Generalised temporal verification diagrams. In *15th Conference on the Foundations of Software Technology and Theoretical Computer Science*, vol. 1026 of *Lecture Notes in Computer Science*, pages 484–498. Springer-Verlag, 1995.

[FMS98] Finkbeiner, B., Manna, Z. and Sipma, H. B.: Deductive verification of modular systems. In W. P. de Roever, H. Langmaack and A. Pnueli, (eds). *Compositionality: The Significant Difference, COMPOS'97*, vol. 1536 of *Lecture Notes in Computer Science*, pages 239–275. Springer-Verlag, December 1998.

[MBB99] Manna, Z., Bjørner, N. S., Browne, A., Colón, M., Finkbeiner, B., Pichora, M., Sipma, H. B. and Uribe, T. E.: An update on STeP: Deductive-algorithmic verification of reactive systems. In R. Berghammer and Y. Lakhnech, (eds). *Tool Support for System Specification, Development and Verification*, Advances in Computing Science, pages 174–188. Springer-Verlag, 1999.

[MBS98] Manna, Z., Browne, A., Sipma, H. B. and Uribe, T. E.: Visual abstractions for temporal verification. In A. Haeberer, (ed.). *Algebraic Methodology and Software Technology (AMAST'98)*, vol. 1548 of *Lecture Notes in Computer Science*, pages 28–41. Springer-Verlag, December 1998.

[MP94] Manna, Z. and Pnueli, A.: Temporal verification diagrams. In M. Hagiya and J. C. Mitchell, (eds). *Proc. International Symposium on Theoretical Aspects of Computer Software*, vol. 789 of *Lecture Notes in Computer Science*, pages 726–765. Springer-Verlag, 1994.

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

Received February 2000
Accepted in revised form December 2000