

JM

REPORT DOCUMENTATION PAGE

Form Approved

OMB NO. 0704-0188

Public Reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188,) Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE 11/30/01	3. REPORT TYPE AND DATES COVERED Final Progress report, 8/1/98 – 7/31/01	
4. TITLE AND SUBTITLE: Abstraction and Compositionality for the Verification of Infinite-State Reactive Systems			5. FUNDING NUMBERS: DAAG 55-98-1-0471	
6. AUTHOR(S): Prof. Zohar Manna and Dr. Henny Sipma				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES): Stanford University Computer Science Department Stanford CA 94305-9045			8. PERFORMING ORGANIZATION: Stanford University REPORT NUMBER: none	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES): U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARO 38760.1-C1	
11. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.				
12 a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12 b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <i>see attached</i>				
14. SUBJECT TERMS: verification, abstraction, compositionality, reactive systems, diagram verification			15. NUMBER OF PAGES: 7	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION ON THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

20020201 136

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

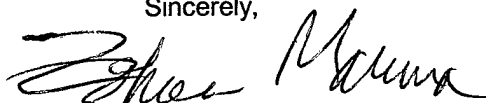
Reprint (Orig + 2 copies)
Manuscript (1 copy)

Technical Report (Orig + 2 copies)
Final Progress Report (Orig + 2 copies)
Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: DAAG-55-9801-0471

REPORT TITLE: Final Report for Abstraction and Compositionality for the Verification of Infinite-State Reactive Systems is forwarded for your information.

Sincerely,

A handwritten signature in black ink, appearing to read "Zohar Manna". The signature is fluid and cursive, with the first name "Zohar" and the last name "Manna" clearly distinguishable.

Zohar Manna, Professor and P.I.

DAAG 55-98-1-0471 Abstract

We describe several techniques for verifying infinite-state systems via finite-state abstractions. Diagrams are top-down property-driven abstractions, which are especially suited for compositional, assume-guarantee reasoning. Predicate abstraction uses a bottom-up approach for generating abstractions; invariant generation techniques are applied to automatically generate the required predicates. Extended finite-state abstractions allow inclusion of extra information produced by the deductive abstraction, which can be used by the model checker to reduce the number of spurious counterexamples.

These methods have been or currently are being implemented in the Stanford Temporal Prover. The methods have been applied in the analysis of a medical device.

Final Progress Report
ARO Contract DAAG 55-98-1-0471
November 30, 2001

P.I.: Prof. Zohar Manna
Computer Science Department
Stanford University
Stanford, CA. 94305-9045

Project Title: Abstraction and Compositionality for the Verification of Infinite-State Reactive Systems

Problem Statement

Software systems are usually *infinite-state*, since they contain system variables over unbounded domains, such as integers, lists, trees, and other data types. Most finite-state verification methods, such as model checking, cannot be applied directly to such systems. The application of temporal verification techniques to software systems is further limited by the size and complexity of the systems analyzed.

Deductive verification, which relies on general theorem-proving and user interaction, provides complete proof systems that can, in principle, prove the correctness of any property over an infinite-state system, provided the property is indeed valid for that system. However, these methods are also limited by the size and complexity of the system being analyzed, becoming much more laborious as the system complexity grows.

Verification methods analogous to those used to manage complexity in software design can be used to overcome these limitations. *Modular verification* follows the classic divide-and-conquer paradigm, where portions of a complex system are analyzed independently of each other. It holds the promise of proof reuse and the creation of libraries of verified components. *Abstraction* is based on ignoring details as much as possible, often simplifying the domain of computation of the original system. This may allow, for

instance, abstracting infinite-state systems to finite-state ones that can be more easily model checked.

Summary of Results

Diagram Verification

Diagrams are property-driven abstractions of a system: verification is only concerned with those aspects of the program that are directly related to the property, thus reducing the burden on the user. The theory of diagrams and their application in the verification of reactive, real-time and hybrid systems is described in [Sip99].

Diagrams can be applied compositionally. Diagrams are constructed and justified for each component individually, taking into account environment assumptions and restrictions. Being automata-based, these diagrams can then be composed by taking products of automata, automatically discharging the assumptions, again justified by first-order verification conditions.

Diagrams can also be used to prove safety properties of parameterized systems, that is, systems that consist of an unspecified number of identical components that interact with each other. To prove liveness properties of parameterized systems we developed the technique of dynamic induction on diagrams[MS99], which allows the verification of the property for a single component to be used to infer the validity of the property for the global system, under the appropriate ordering conditions.

Automatic generation of diagrams is hampered by the fact that the starting diagram, the automaton for the property to be proven, is exponential in the size of the formula. To alleviate this problem, we explored the use of alternating automata, which are linear in the size of the formula. In [MS00] we demonstrated the use of alternating automata in the deductive verification of safety properties. We are currently extending this to the deductive verification of progress properties. Although generally applicable, this method appears to be especially suitable for assume-guarantee properties.

Program Abstraction by Invariant Generation

In [CU98] we presented a two-phase approach to program abstraction. It first uses theorem proving to construct a finite-state abstraction of an infinite-state program, and then finite-state analysis to compute the reachable states of the abstraction. This set of reachable abstract states is then used to verify

temporal properties of the concrete system. This method, while highly automated, requires user guidance in the form of a finite set of atomic assertions over the variables of the concrete program.

Invariant generation can be used to generate these abstractions automatically. We use the decidable theory of linear inequalities as a basis to discover program invariants. Our approach is to symbolically simulate the program for a number of program steps, representing them by linear systems, and then search for invariants among the common consequences of these systems. The advantage of this deductive variant of linear invariant generation is its generalizability: it admits the presence of disequalities and strict inequalities, thereby enabling the generation of more precise invariants in, for example, the branches of conditional statements.

We have also used this technique to automatically generate ranking functions for establishing loop termination [CS01]. The technique reduces the search for linear ranking functions to the problem of finding certain consequences of two linear systems – one approximating the transition relation around the loop and the other approximating the states reachable while in the loop. By manipulating these systems, the algorithm isolates those consequences that define linear ranking functions.

Extended Finite-state Abstraction

Many deductive and deductive-algorithmic verification methods explicitly or implicitly construct finite-state system abstractions, which are explicitly or implicitly model checked. We show how such abstractions can be represented, combined and model checked in a general way. For this, we define a class of extended finite-state abstractions, and present an algorithmic model checking procedure for them. This procedure uses all the information produced by the deductive algorithmic methods, in a finite-state format that can be easily and incrementally combined. Besides a standard \forall CTL*-preserving safety component, the extended abstractions include extra bounds on fair transitions, well-founded orders, and constrained transition relations for checking existential properties or the generation of LTL counterexamples. This approach minimizes the need for user interaction and maximizes the impact of the available automated deduction and model checking tools. Once proved, verification conditions are re-used as much as possible, leaving the temporal and combinatorial reasoning to automatic tools. The method is described in detail in [MSU99, Uri00, Uri01].

Applications

In the final year of the contract we have started to apply above methods to the analysis of a medical device: a computer-assisted resuscitation device developed by the Walter Reed Army Institute of Research (WRAIR). Based on detailed tagged requirements developed by physicians at WRAIR, a clocked transition system was created to model the system. The system, consisting of some 400 transitions, was divided into modules, interacting by shared variables, and provided with environment assumptions both on timing behavior and data modification. Abstraction techniques and modular reasoning were used to check the system for infinite loops. Further analysis of this system is planned when funding is secured.

Implementation

The modular verification techniques proposed in [FMS98, BMSU01] were implemented in STeP (Stanford Temporal Prover) for reactive and real-time systems. We are currently implementing these methods for hybrid systems. An overview of the STeP system can be found in [BBC⁺00].

Except for the first-order theorem-proving component, STeP has been reimplemented in Java, with the objective to obtain a more modular architecture that is easily extensible with new methods and computational models. It allows for quick experimentation to evaluate new techniques.

Publications

Journal papers

- Nikolaaj S. Bjørner and Anca Browne, Michael Colón, Bernd Finkbeiner, Zohar Manna, Henny B. Sipma, Tomás E. Uribe, Verifying Temporal Properties of Reactive Systems: A STeP Tutorial. *Formal Methods in System Design*, Vol 16, June 2000.

Conference papers

- Anca Browne, Henny Sipma, and Ting Zhang, Linking STeP with SPIN. In *SPIN Model Checking and Software Verification, 7th International SPIN Workshop*, vol 1885 of *Lecture Notes in Computer Science*, pages 181–186. Springer Verlag 2000.

- Michael Colon and Henny Sipma, Synthesis of Linear Ranking Functions. In *Proceedings 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol 2031 of *Lecture Notes in Computer Science*, pages 67–81. Springer Verlag 2001.
- Bernd Finkbeiner, Language Containment Checking with Nondeterministic BDD's. In *Proceedings 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol 2031 of *Lecture Notes in Computer Science*, pages 24–38. Springer Verlag 2001.
- Bernd Finkbeiner and Henny Sipma, Checking Finite Traces using Alternating Automata. In *Electronic Notes in Theoretical Computer Science*, volume 55, no 2. Elsevier Science Publishers, 2001.
- Zohar Manna, Nikolaj Bjorner, Anca Browne, Michael Colon, Bernd Finkbeiner, Mark Pichora, Henny Sipma, and Tomas Uribe, An Update on STeP: Deductive-Algorithmic Verification of Reactive Systems. In *Tool Support for System Specification, Development and Verification*, Advances in Computing Science, pages 174–188. Springer Verlag, 1999.
- Zohar Manna, Anca Browne, Henny Sipma, and Tomas Uribe, Visual Abstractions for Temporal Verification. In *Proceedings Algebraic Methodology and Software Technology*, vol 1548 of *Lecture Notes in Computer Science*, pages 28–41. Springer Verlag 1998.
- Zohar Manna and Henny Sipma, Verification of Parameterized Systems by Dynamic Induction on Diagrams. In *Proceedings 11th International Conference on Computer Aided Verification*, vol 1633 of *Lecture Notes in Computer Science*, pages 25–43. Springer Verlag 1999.
- Zohar Manna and Henny Sipma, Alternating the Temporal Picture for Safety. In *Proceedings 27th International Colloquium on Automata and Languages Prog.*, vol 1853 of *Lecture Notes in Computer Science*, pages 429–450. Springer-Verlag, 2000.

List of Scientific Personnel

Scientific personnel who participated in the project:

- **Faculty:** Prof. Zohar Manna
- **Scientific Personnel:** Dr. Henny B. Sipma, Anca Browne
- **Graduate students:** Michael Colon, Bernd Finkbeiner, Henny Sipma, Tomas Uribe, Ting Zhang
- **Visitors:** Prof. Saddek Bensalem, VERIMAG, France; Arnab Ray, SUNY Stony Brook.

Graduated Ph.D. students:

- Henny Sipma February 1999. Thesis: *Diagram-Based verification of reactive, real-time and hybrid systems*
- Tomás E. Uribe, December 1998. Thesis: *Abstraction-based Deductive-Algorithmic Verification of Reactive Systems.*

References

- [BBC⁺00] Nikolaj S. Bjørner, Anca Browne, Michael Colón, Bernd Finkbeiner, Zohar Manna, Henny B. Sipma, and Tomás E. Uribe. Verifying temporal properties of reactive systems: A STeP tutorial. *Formal Methods in System Design*, 16(3):227–270, June 2000.
- [BMSU01] Nikolaj S. Bjørner, Zohar Manna, Henny B. Sipma, and Tomás E. Uribe. Deductive verification of real-time systems using STeP. *Theoretical Computer Science*, 253:27–60, 2001.
- [CS01] Michael Colon and Henny Sipma. Synthesis of linear ranking functions. In Tiziana Margaria and Wang Yi, editors, *7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, LNCS. Springer Verlag, April 2001. To appear.
- [CU98] Michael A. Colón and Tomás E. Uribe. Generating finite-state abstractions of reactive systems using decision procedures. In

- Alan J. Hu and Moshe Y. Vardi, editors, *Proc. 10th Intl. Conference on Computer Aided Verification*, volume 1427 of *LNCS*, pages 293–304. Springer-Verlag, July 1998.
- [FMS98] Bernd Finkbeiner, Zohar Manna, and Henny B. Sipma. Deductive verification of modular systems. In Willem-Paul de Roever, Hans Langmaack, and Amir Pnueli, editors, *Compositionality: The Significant Difference, COMPOS'97*, volume 1536 of *LNCS*, pages 239–275. Springer-Verlag, December 1998.
- [MS99] Zohar Manna and Henny B. Sipma. Verification of parameterized systems by dynamic induction on diagrams. In Nicholas Halbwachs and Doron Peled, editors, *Proc. 11th Intl. Conference on Computer Aided Verification*, volume 1633 of *LNCS*, pages 25–43, Trento, Italy, July 1999. Springer-Verlag.
- [MS00] Zohar Manna and Henny B. Sipma. Alternating the temporal picture for safety. In Ugo Montanari, Jose D.P. Rolim, and Emo Welzl, editors, *Proc. 27th Intl. Colloq. Aut. Lang. Prog.*, volume 1853, pages 429–450, Geneva, Switzerland, July 2000. Springer-Verlag.
- [MSU99] Zohar Manna, Henny B. Sipma, and Tomás E. Uribe. Deductive model checking and abstraction. In *Monterey Workshop on Engineering Automation for Computer Based Systems*, pages 77–85, April 1999.
- [Sip99] Henny B. Sipma. *Diagram-based Verification of Discrete, Real-time and Hybrid Systems*. PhD thesis, Computer Science Department, Stanford University, February 1999.
- [Uri00] Tomás E. Uribe. Combinations of model checking and theorem proving. In *FroCos'2000, 3rd International Workshop on Frontiers of Combining Systems*, volume 1794 of *Lecture Notes in Artificial Intelligence*, pages 151–170, Nancy, France, March 2000. Springer-Verlag.
- [Uri01] Tomas Uribe. Model checking extended finite-state abstractions. *Science of Computer Programming*, 2001.