https://doi.org/10.1007/s00165-019-00500-4
BCS © 2019
Formal Aspects of Computing (2019) 31: 643–660

Formal Aspects
of Computing

Zohar Manna (1939–2018)

Nachum Dershowitz[1] and Richard Waldinger[2]

[1]School of Computer Science, Tel Aviv University, Tel Aviv-Yafo, Israel
[2]Artificial Intelligence Center, SRI International, Menlo Park, CA, USA

ולא היה בינינו אלא זהר

[And what we shared was solely
zohar.] —Leah Goldberg (1944)

Zohar Manna, founding father of the study and application of formal methods for software and hardware verification, died peacefully at his home in Netanya, Israel, a year ago, on August 30, 2018, after a long and marvelously productive career. He passed away surrounded by his extended family. He was 79.

Zohar, whose name in Hebrew means "illumination," was born in early 1939 and grew up in Haifa, Israel. He met his future wife, Nitza Kletchevsky, at a folk-dancing class in Haifa in 1959. The couple married in 1963. Zohar is survived by Nitza and their children—daughters, Yonit, a software engineer, Hagit, a psychiatrist, Irit, an operations manager, and son Amit, a product manager—all of whom were at his bedside when he passed away. He also left five grandchildren, three boys and two girls.

Correspondence to: Nachum Dershowitz, E-mail: nachumd@tauex.tau.ac.il

Photo courtesy Manna family.

1. Zohar's academic life

Zohar received his bachelor's and master's degrees in mathematics from the Technion in Haifa in 1961 and 1965, respectively, serving as a scientific programmer, with the rank of First Lieutenant, in the Israel Defense Forces from 1962 to 1964. Afterwards, he attended Carnegie Mellon University in Pittsburgh (together with one of us, Richard), where he earned his Ph.D. in computer science in the spring of 1968 under the guidance of Robert W Floyd and Alan J. Perlis.

Going backwards, we find that his advisor, Alan Perlis, was an academic descendant of a long line of mathematicians and scientists,[1] starting with
Philip Franklin, who was a student of
Oswald Veblen, who was a student of
Eliakim Hastings Moore, whose doctorvater was
Hubert Anson Newton, who was a student of
Michel Chasles, who was a student of
Siméon-Denis Poisson, who was a student of both
Joseph-Louis Lagrange (Giuseppe Luigi Lagrangia), who studied under
Leonhard Euler—but had "no dissertation, no advisor,"[2]
and of
Pierre-Simon Laplace (Peter Simon de la Place), who was a student of
Jean-Baptiste le Rond d'Alembert,
—who received his baccalauréat en arts from
Collège des Quatre-Nations (Collège Mazarin) in 1735.
Bob Floyd had done his undergraduate studies in Chicago, but did not obtain a doctorate.

Upon graduating CMU, Zohar joined the Stanford University faculty as an assistant professor of computer science, where, among other activities, he worked with artificial intelligence pioneer John McCarthy. Zohar returned to Israel in 1972 to the Department of Applied Mathematics at the Weizmann Institute of Science in Rehovot, first as an associate professor and from 1976 on as full professor (where he supervised the dissertation of the other one of us, Nachum). In 1975, he was recruited back to Stanford (taking Nachum along with him), becoming a full professor of computer science in 1978, and dividing his time between Stanford and Weizmann until 1995, at which point he resigned the latter appointment. He spent many years at the Stanford AI Lab (SAIL), located on Arastradero Road in the Palo Alto foothills, where a unique atmosphere of playful experimentation and electrifying discovery permeated the halls. He remained at Stanford University until his retirement in 2010.

Over a career spanning nearly half a century, Zohar had profound impact on most aspects of formal methods and automated reasoning. His work inspired several generations of researchers. He pioneered programme verification and programme synthesis, two fields that were then at the theoretical edge of computing, but which today help form the foundations of artificial intelligence and help assure the reliability of extraordinarily complex software.

Zohar trained 30 Ph.D. students, many of whom went on to top academic posts in the United States, Europe, and Israel (see Sect. 3). He authored or coauthored 9 books and close to 200 scholarly articles. A list of most of Zohar's many publications appears at the end of this article. This list includes more than 50 technical reports (Weizmann, Stanford, SRI), which served as a rapid mode of dissemination of pre-publication cutting-edge research.

Zohar was universally acclaimed and deeply appreciated as a consummate teacher. Even as a graduate student, his first talk, describing his thesis research, was polished and professional. The courses he taught include "Logic and Automated Reasoning" and "Formal Methods for Concurrent and Reactive Systems." In those pre-PowerPoint days, when academics used gesticulations and blackboards to illustrate their points, Zohar developed a reputation as an engaging and energetic lecturer. His friends called him the "fastest chalk in the West." Moshe Vardi divulged that "His course was one of my favorite all-time computer science courses. It convinced me that logic has a real place in computer science."

[1] See https://www.genealogy.math.ndsu.nodak.edu/id.php?id=19351 and follow the links backwards in history.

[2] Euler's pedigree on the Mathematics Genealogy Project (https://www.genealogy.math.ndsu.nodak.edu/id.php?id=38586) goes back another 25 generations, through Desiderius Erasmus until Sharaf al-Dīn al-Ṭūsī of twelfth century Persia.

His international cachet is reflected in the honors he accrued, including:

- the ACM Programming Systems and Languages Award (1974, for [38]);
- a Guggenheim Fellowship (1981);
- the F. L. Bauer Prize from the Technical University Munich (1992);
- Fellowship in the Association for Computing Machinery (1993, the first year awarded);
- *doctor honoris causa* from École normale supérieure de Cachan (France, 2002);
- and a Fulbright Fellowship (2002).

In 2016, he (together with Richard Waldinger) received the Herbrand Award for Distinguished Contributions to Automated Reasoning from CADE, Inc. (parent organisation of the International Conference on Automated Deduction series) "for his pioneering research and pedagogical contributions to automated reasoning, programme synthesis, planning, and formal methods." He was also associate editor of *Acta Informatica* and of *Theoretical Computer Science* and a board member of the International Institute for Software Technology of the United Nations University in Macao.

Zohar wove a world wide web of friends and colleagues. His fifty coauthors and collaborators read like a computer-science hall of fame. Besides his students, these include Stephen Boyd (Stanford University), Manfred Broy (Technical University Munich), Domenico Cantone (University of Catania), Hubert Comon (ENS Cachan), David Dill (Stanford University), John McCarthy (Stanford University), Peter Pepper (University of Berlin), the late Amir Pnueli (Tel Aviv University, New York University, and Weizmann Institute), Alberto Policriti (University of Udine), Shankar Sastry (UC Berkeley), the late Mark Stickel (SRI International), Claire Tomlin (Stanford University), and Richard Waldinger (SRI International).

The nine books bearing Zohar's authorship are all models of clarity and comprehensiveness:

1. *Mathematical Theory of Computation*, McGraw-Hill, New York, NY, 1974 [39]. This classic has been translated into Bulgarian, Czech, Hungarian, Italian, Japanese, and Russian. Reprinted in a Dover edition, 2003.
2. *Studies in Automatic Programming Logic*, American-Elsevier, New York, NY, 1977, with Richard Waldinger [66].
3. *Lectures on the Logic of Computer Programming*, CBMS-NSF Regional Conference Series in Applied Mathematics, No. 31, SIAM, 1980 [97].
4. *Logical Basis for Computer Programming*, Volume 1: Deductive Reasoning, Addison-Wesley, Reading, MA, 1985, with Richard Waldinger [143].
5. *Logical Basis for Computer Programming*, Volume 2: Deductive Systems, Addison-Wesley, Reading, MA, 1990, with Richard Waldinger [172].
6. *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer-Verlag, New York, 1992, with Amir Pnueli [190].
7. *The Deductive Foundations of Computer Programming*, Addison-Wesley, Reading, MA, 1993, with Richard Waldinger [125].
8. *Temporal Verification of Reactive Systems: Safety*, Springer-Verlag, New York, 1995, with Amir Pnueli [219].
9. *The Calculus of Computation: Decision Procedures with Applications to Verification*, Springer-Verlag, Berlin, 1998, with his student Aaron Bradley [292].

See Fig. 1.

A third volume (following items 6 and 8 in the above list) of his magisterial series with Pnueli on reactive system, *The Temporal Logic of Reactive and Concurrent Systems: Progress*, remained unfinished at the time of Amir's untimely passing in 2009.[3]

[3] The first chapter was eventually (2010) published in a volume in memory of Amir [300]. Two additional chapters are available at http://theory.stanford.edu/~zm/tvors3.html.

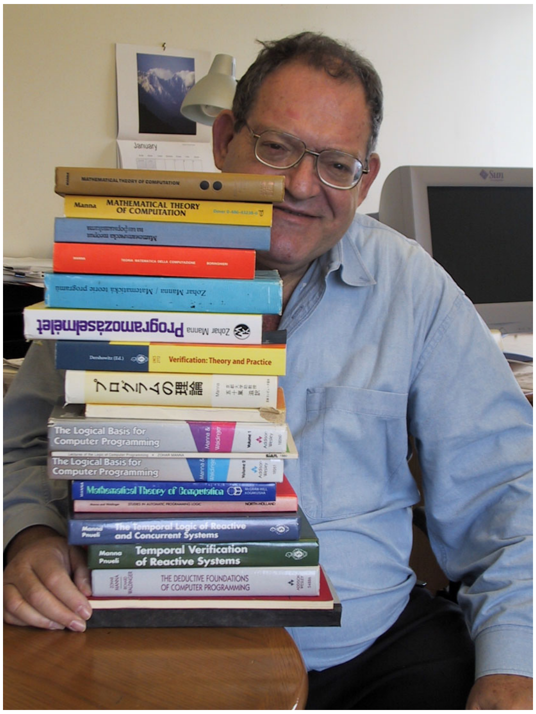
MATHEMATICAL THEORY OF COMPUTATION
Manna MATHEMATICAL THEORY OF COMPUTATION
Zohar Manna Programozáselmélet
Verification: Theory and Practice
The Logical Basis for Computer Programming
The Logical Basis for Computer Programming
Mathematical Theory of Computation
The Temporal Logic of Reactive and Concurrent Systems
Manna Pnueli Temporal Verification of Reactive Systems
THE DEDUCTIVE FOUNDATIONS OF COMPUTER PROGRAMMING

Fig. 1. Zohar behind a pile of his books and the festschrift in his honor (February 2004; courtesy Matteo Slanina).

Zohar supervised the design and development of three software systems:

- the *Tableau Deductive System* (1990), distributed by Chariot Software Group;
- *STeP: Stanford Temporal Prover* (1995), a verification system for reactive and concurrent systems, which was distributed as educational freeware; and
- πVC (2007), a verifying compiler to accompany the textbook, *The Calculus of Computation*.

When Zohar turned sixty-four in 2003, an age of numerological significance to cognoscenti in the realm of binary computers,[4] many of his graduate students, research collaborators, and computer-science colleagues gathered in Sicily for a symposium on subjects related to Zohar's manifold contributions in the field. Their breadth and depth serve as a tribute to Zohar's lasting impact on the field. The symposium was held in Taormina between June 29 and July 4, 2003. There were lectures by Martín Abadi, Luca de Alfaro, Rajeev Alur, Dines Bjørner, Egon Börger, Manfred Broy, Domenico Cantone, the late Ashok Chandra, Hubert Comon, Patrick Cousot, Willem-Paul de Roever, Nachum Dershowitz, Alfredo Ferro, Bernd Finkbeiner, Thomas Henzinger, Gérard Huet, Shmuel Katz, Jean-Louis Lassez, Tom Maibaum, Ugo Montanari, Ben Moszkowski, Krishna Palem, Peter Pepper, the late Amir Pnueli, Alberto Policriti, and Jean Vuillemin.

Another, one-day symposium in Zohar's honor was held a few weeks earlier at Tel Aviv University on May 16, 2003, with talks by Yaacov Choueka, Nachum Dershowitz, Danny Dolev, Amos Fiat, Orna Grumberg, David Harel, Adi Shamir, and Haim Wolfson. In addition, a liber amicorum was prepared in tribute: *Verification – Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday* (N. Dershowitz, ed., Springer, 2003).

[4] The choice of the $1{,}000{,}000_2$-th birthday in computer circles dates back at least to the honoring of John McCarthy's birthday in 1991, conceived by Don Knuth and Jeff Ullman. It is also the subject of the famous Beatles number, "When I'm Sixty-Four," by John Lennon and Paul McCartney, recorded in 1966, but composed earlier.

2. Zohar's research

Zohar was a deep thinker who laid the foundations for tools that are now coming into widespread use. His manifold research interests covered much of the theoretical and practical aspects of the logical half of computer science, embracing all of the following: the theory of computation; logics of programmes; automated deduction; concurrent and reactive systems; real-time and hybrid systems; verification of hardware and software; and synthesis of programmes. His students and colleagues dedicated their research careers to the hardest problems in automated reasoning, including programme semantics, partial correctness, termination, invariant generation, programme synthesis, programme transformation, planning, proof methodology, temporal reasoning, natural-language understanding, nonclausal proof search, and decision procedures. Each of these activities is today a thriving independent field of research.

Zohar's 1968 dissertation, entitled *Termination of Programs* [1], used first-order logic to formalise the problem of termination for abstract programmes and demonstrate decidable subclasses of the termination programme. His advisors, Alan Perlis and Bob Floyd, were both Turing Award recipients. The abstract reads as follows:

> The thesis contains two parts which are self-contained units.
> In Part I we present several results on the relation between (1) the problem of termination and equivalence of programs and abstract programs, and (2) the first-order predicate calculus.
> Part II is concerned with the relation between (1) the termination of interpreted graphs, and (2) properties of well-ordered sets and graph theory.

Whereas Floyd's own approach to termination required the invention of inductive assertions and well-founded orderings, Zohar's approach required none of these. Richard Lipton calls it "one of the coolest PhD theses ever."[5]

Zohar's magnificent textbook, *Mathematical Theory of Computation* [39], was extraordinarily influential. It pioneered the logical analysis of programmes for correctness vis-à-specifications and for termination properties. For very many of today's computer scientists, this book was their introduction to formal methods. An example of Zohar's unsung generosity is this acknowledgment by Marin Vlada (now a professor of computer science at the University of Bucharest):[6]

> I must also thank Professor Manna who in 1977 sent me – by package post – his *Mathematical Theory of Computation*, published in 1974 [and several scientific articles]. I was a student, and I used everything he sent me to advance my undergraduate work. I must admit that I was very surprised by his kindness, considering that Romania is separated from the U.S.A. by a very large ocean. . . . Thank you very much sir!

Formal proofs have become increasingly necessary in our software age. The reason is the necessity for assuring the safety of complex modern applications, which often involve what scientists call the "concurrent interaction" of many different subprogrammes and hardware routines. These runtime interactions have increased to a point that it's impossible to adequately test all the potential cross-influences using traditional methods.

Zohar played a central role in the study of applications of temporal logic, in particular, to the analysis of complex software systems, especially of the concurrent and reactive varieties. And, indeed, his most cited works are on the formal analysis of reactive systems, much of which was done in collaboration with Amir Pnueli. In the early seventies, they showed how first-order logic could be used to reason about functional programmes [11]; this line of work was continued with various coauthors (for example, [38]). Later in that decade, they embarked on their seminal study of applications of temporal modal logics to verification of concurrent programmes, followed by work with their students on the formal specification and verification of timed and hybrid systems. Their long-term collaboration culminated in two landmark volumes [190, 219] and in the *STeP* prover. All told, Zohar and Amir published some 45 joint works.

One of Zohar's most highly cited papers (with Nachum Dershowitz, 1979) is "Proving Termination with Multiset Orderings" [91] in the *Communications of the ACM*. Their well-founded ordering on bags of values turned out to be extremely powerful and useful in difficult termination arguments.

[5] https://rjlipton.wordpress.com/2014/11/12/three-in-the-room.

[6] http://mvlada.blogspot.com/2011/03/professor-zohar-manna.html.

Richard Waldinger was Zohar's chief collaborator on programme synthesis. In 1970, they published a groundbreaking paper, "Toward automatic program synthesis," also in the *Communications of the ACM* [17]. Later, in "Synthesis: Dreams ⇒ Programs" [88] they employed programme transformations to deal with imperative as well as functional programmes within a system dubbed DEDALUS. Eventually they moved to a purely "deductive" approach in which, to construct a programme meeting a desired condition for a given input, one proves the existence of an output entity that satisfies that condition. The proof is restricted to be sufficiently constructive so that a programme that computes the satisfying output can be extracted. Conditional expressions are introduced via case analysis in the proof; recursion is introduced by application of *recursion induction*.

Perhaps the most practical application of deductive programme synthesis came from its use by NASA in the Amphion project,[7] which synthesised programmes for analysis of data from interplanetary missions. Amphion began from a graphical specification elicited interactively via a dialogue with the user, a planetary scientist, typically with no knowledge of logic or programming. The graph was translated into logical form and submitted to a theorem prover equipped with axioms that described subroutines from a JPL library. A straight-line programme that met the specification, composed of routines from the library, was extracted from the proof. That programme was translated into Fortran and presented to the user. programmes synthesised by Amphion were used to analyse photographs from the Cassini mission to Saturn.

Zohar and Richard also developed a nonclausal reasoner that better served the need for doing an inductive proof in a resolution-theorem-proving framework. This collaboration culminated in the two-volume book, *The Logical Basis for Computer Programming* [143, 172] (1985 and 1990).

In the mid-seventies, Zohar was impressed with a paper by Rod Burstall, "Program Proving as Hand Simulation with a Little Induction" (*IFIP Congress*, Stockholm, Aug. 1974). The method seemed to provide a sometimes superior alternative to the Floyd/Naur/Hoare invariant assertion method, which dominated the field. While the invariant assertion required affixing propositions to a programme that were claimed to be true whenever control passed through the corresponding point, Burstall's method allowed propositions that were true at some of those times, but not necessarily all. It turned out that the assertions required by the Burstall method were oftentimes simpler than invariant assertions. In a 1978 paper with Waldinger, "Is 'sometime' Sometimes Better than 'always'?: Intermittent Assertions in Proving Program Correctness" [80], Zohar named the approach the *intermittent assertion* method, and pointed out its advantages. (The paper contributed to the subsequent interest in temporal logic in the formal methods community.)

In response, Edsger Dijkstra wrote this to Zohar:[8]

> Last week I spoke to one of the attendants of the recent Summer School organized by the Mathematical Centre in Amsterdam, at which you were one of the lecturers. As far as he was concerned you have stolen the show: he was most impressed by your performances and I thought that you might like to hear so. . . .
> You and Richard Waldinger deserve each reader's compliments for the way in which your report has been phrased: for that very reason it was a pleasure and a privilege to read it. I think that I understood every sentence of it, something that is in sharp contrast to what is usually dropped in my mailbox. . . . Your report is exceptionally well-written. Thank you, it is nice to see indispensable standards sometimes maintained!

3. Zohar's academic progeny

From the point of view of the success of academic descendants in computer science, Zohar ranked #2 in the list of "The Best Nurturers in Computer Science Research."[9] He took immense pride in his students and their spiritual heirs. All told, he supervised thirty doctoral students (and has close to 300 Ph.D. descendants to date):[10]

1. Jean-Marie Cadiou (Stanford University, 1972): *Recursive Definitions of Partial Functions and their Computations*; Entymologist; Director, Institute for the Protection and the Security of the Citizen, Joint Research Center (passed away May 2007)
2. Ashok Chandra (Stanford University, 1973): *On the Properties and Applications of Program Schemas*; General Manager, Microsoft Research, Mountain View, CA (passed away November 2014)

[7] See https://ti.arc.nasa.gov/tech/rse/synthesis-projects-applications/pro-amphion.

[8] https://www.cs.utexas.edu/users/EWD/ewd05xx/EWD574.PDF (July 26, 1976).

[9] Bharath Kumar M. and Y. N. Srikant, *Proceedings of the 2005 SIAM International Conference on Data Mining*, https://epubs.siam.org/doi/pdf/10.1137/1.9781611972757.62. Their criteria embody the notion that "The gurus with the highest *gurudakshina* are the best nurturers."

[10] See https://www.genealogy.math.ndsu.nodak.edu/id.php?id=19351 for subsequent generations.

3. Jean Vuillemin (Stanford University, 1973): *Proof Techniques for Recursive Programs*; Professor, Computer Science, École normale supérieure
4. Shmuel Katz (Weizmann Institute, 1976): *Invariants and the Logical Analysis of Programs*; Professor, Computer Science, Technion – Israel Institute of Technology
5. Adi Shamir (Weizmann Institute, 1977): *Fixedpoints of Recursive Programs*; Professor, Applied Mathematics and Computer Science, Weizmann Institute of Science
6. Nachum Dershowitz (Weizmann Institute, 1979): *Automatic Program Modification*; Professor, Computer Science, Tel Aviv University
7. William L. Scherlis (Stanford University, 1980): *Expression Procedures and Program Derivation*; Director, Institute for Software Research, Carnegie Mellon University
8. Pierre Wolper (Stanford University, 1982): *Synthesis of Communicating Processes from Temporal Logic Specifications*; Rector, University of Liège
9. Ben Moszkowski (Stanford University, 1983): *Reasoning about Digital Circuits*; Senior Research Fellow, Software Technology Research Laboratory, De Montfort University
10. Yoni Malachi (Stanford University, 1985): *Nonclausal Logic Programming*; Vice President, Research & Development, Radvision (passed away April 2004)
11. Martín Abadi (Stanford University, 1986): *Temporal Theorem Proving*; Professor Emeritus, Computer Science, University of California Santa Cruz; Google
12. Marianne Baudinet (Stanford University, 1988): *Logic Programming Semantics: Techniques and Applications*; Information Technology Director, Euroclear
13. Rajeev Alur (Stanford University, 1991): *Techniques for Automatic Verification of Real-Time Systems*; Professor, Computer and Information Science, University of Pennsylvania
14. Thomas A. Henzinger (Stanford University, 1991): *The Temporal Specification and Verification of Real-Time Systems*; President, Institute of Science and Technology Austria
15. Eddie Chang (Stanford University, 1994): *Compositional Verification of Reactive and Real-Time Systems*; former Federal Prosecutor, Southern District of New York and District of Connecticut; Second Vice President, Cyber Risk Management, Travelers
16. Hugh McGuire (Stanford University, 1995): *Two Methods for Checking Formulas of Temporal Logic*; Embedded Software Consultant, Green Hills Software
17. Anuchit Anuchitanukul (Stanford University, 1995): *Synthesis of Reactive Programs*; First Executive Vice President, Kiatnakin Bank
18. Arjun Kapur (Stanford University, 1997): *Interval and Point-Based Approaches to Hybrid System Verification*; Vice President and General Manager, Imaging and Camera Technologies Group, Intel
19. Luca de Alfaro (Stanford University, 1998): *Formal Verification of Probabilistic Systems*; Professor, Computer Science, University of California Santa Cruz
20. Nikolaj S. Bjørner (Stanford University, 1998): *Integrating Decision Procedures for Temporal Verification*; Principal Researcher, Microsoft Research
21. Tomás E. Uribe (Stanford University, 1998): *Abstraction-Based Deductive-Algorithmic Verification of Reactive Systems*; Senior Computer Scientist, RelationalAI
22. Henny B. Sipma (Stanford University, 1999): *Diagram-Based Verification of Discrete, Real-Time and Hybrid Systems*; Chief Technology Officer, Kestrel Technology
23. Bernd E. Finkbeiner (Stanford University, 2002): *Verification Algorithms Based on Alternating Automata*; Professor, Computer Science, Universität des Saarlandes
24. Michael Colón (Stanford University, 2003): *Deductive Techniques for Program Analysis*; Computer Scientist, US Naval Research Laboratory
25. Calogero Zarba (Stanford University, 2004): *The Combination Problem in Automated Reasoning*; Software Developer, Neodata Group
26. Sriram Sankaranarayanan (Stanford University, 2005): *Mathematical Analysis of Programs*; Associate Professor, Computer Science, University of Colorado
27. Ting Zhang (Stanford University, 2006): *Arithmetic Integration of Decision Procedures*; Assistant Professor, Computer Science, Iowa State University

28. Matteo Slanina (Stanford University, 2007): *Deductive Verification of Alternating Systems*; Google
29. César Sánchez (Stanford University, 2007): *Deadlock Avoidance for Distributed Real-Time and Embedded Systems*; Associate Research Professor, IMDEA Software Institute
30. Aaron Bradley (Stanford University, 2007): *Safety Analysis of Systems*; Principal Engineer, Mentor Graphics

Among many other contributions, Zohar and his students developed powerful methods for theory of computation (Cadiou, Chandra, Vuillemin, Shamir), programme verification (Katz, Chang, Kapur, de Alfaro, Bjørner, Uribe, Sipma, Finkbeiner, Colón, Sankaranarayanan, Slanina, Bradley), programme transformation and synthesis (Dershowitz, Scherlis, Wolper, Anuchitanukul), temporal and real-time logics (Moszkowski, Abadi, Alur, Henzinger, McGuire, Sánchez), logic programming (Malachi, Baudinet), and theorem proving (Zarba, Zhang, Bradley).

Zohar also hosted several postdocs, including Henny Sipma (Kestrel Technology) and Yan Jurski (Université Paris 7).

4. Reminiscences

Zohar and Nitza continued to dance throughout their years together, wherever they were. They raised their four children while travelling regularly from continent to continent. Their children recall a house full of life and guests, and Friday night family dinners, but most of all, they remember the family camping trips. They also toured the world extensively. Zohar and Nitza enjoyed taking everyone camping to nature reserves or to Zohar's favorite, the beach, where he would play hours of backgammon and Rummikub with the kids. There are dozens of albums with Zohar's exquisite photography, another one of his many talents.

A more mathematical talent of Zohar's was counting cards at blackjack, leading to his being blacklisted by some casinos. "He laughed that it was one of his greatest achievements," his son recalls. Zohar's rolling, infectious laughter is not to be forgotten.

One year, Zohar and Nitza took Richard, and his wife Fran, on a trip to the Sinai Peninsula, which at that time was in Israeli territory. While the trip involved strenuous hikes and climbs and Zohar was not a light person, he was always up in front with the leaders. After one particularly hot day, they arrived at a Bedouin village. In the centre of the village was a well with a rope leading down into the cold water. A man pulled up a crate containing bottles of chilled Coca Cola—a small miracle! Zohar was buying and Richard, who *never* drinks Coca Cola, had two.

During graduate school and throughout his research career, Zohar kept up the habit of a long afternoon nap, which enabled him to work late into the night. When working together on one of their early synthesis papers, Zohar kept urging Richard to work later and later. Instead, Richard suggested that they start work at 7 a.m., never dreaming that Zohar would agree. To Richard's surprise, Zohar called his bluff, and for the several weeks working on that paper, he faithfully showed up at the office at seven. Richard himself was not used to these hours, and one afternoon fell fast asleep on the floor. He woke up suddenly to find Don Knuth and the famous German type designer, Hermann Zapf, peering down at him. Zapf was visiting the department and Knuth was giving him a tour. Perhaps Zapf was not familiar with the informal California lifestyle. After that, Zohar installed a large beanbag chair in his office.

Zohar and Richard would divide their time between Richard's office at SRI and Zohar's office at Stanford—at the AI Lab in the hills or, later, at the Margaret Jacks Building. In the early days, Richard was in the habit of bicycling from SRI to the AI Lab at lunch time. One day, Zohar decided he wanted to try this too. But he happened to pick the hottest day of the year, when the temperature reached 105°F. Richard was used to bicycling but Zohar was not; they had to stop and rest at Rossotti's (the Alpine Inn Beer Garden), where they downed two beers each. It is not recorded how productive that afternoon was.

Fig. 2. Zohar (in the middle) and students (Nikolaj Bjørner and Bernd Finkbeiner to the left; Mark Pichora and Michael Colón, plus Zohar's son, Amit, to the right) floating in the Dead Sea (March 1998).

Richard and Zohar had an unusual method for collaborative writing. Richard is left handed and Zohar was right handed. They sat side by side in front of a single pad of (yes) paper. Richard would write the first version of a sentence. While not a native English speaker, Zohar had an uncanny ability to detect ambiguities, grammatical errors, and infelicities of style. Richard would suggest alternatives, Zohar would critique, and so on. Thus, each sentence was the outcome of a small battle. Even on a first draft, Zohar would annotate each sentence with font and formatting directives. Chapter and section headings were also introduced into the draft. Most of these drafts were eventually put into TEX by Richard's former housemate, Evelyn Eldridge-Diaz.

When Nachum took Zohar's course at Weizmann in 1974, *Mathematical Theory of Computation* was in galley proofs, and exercises were assigned from them. Besides laboring to solve the problems, Nachum would take pleasure in proofreading for errors and typos, for which Zohar was quite grateful. (This skill alone was reason enough to take him on as a student.) Later, when preparing their textbooks, Zohar and Richard would give students handouts in which three small errors had been deliberately introduced into each chapter. As part of their homework, students were asked to find these errors. If they found more than the three, so much the better. This worked well as long as Zohar and Richard kept good records of where the errors had been secreted, so they could be removed prior to publication. For several years, students would read successive versions of the manuscript, sprinkled with deliberately introduced errors.

Richard Lipton relates the following story:[11]

> At a STOC, long ago, I [Lipton] saw [Zohar] and started to explain my result: the solution to [a rewrite-rule termination] puzzle. After hearing it he said that he liked the solution and then added that he once had worked on this problem. . . .
>
> It turned out that he and Steve Ness had [an obscure] paper "On the termination of Markov algorithms" at the 1970 Hawaiian International Conference on System Science, held in Honolulu. Zohar explained the conference was not the "best," but was held every year in the Hawaiian Islands. In January. Where it is warm. And sunny. It sounded like a great conference to me. . . .
>
> He also explained to me the "three-person rule." The Hawaiian conference was highly parallel and covered many areas of research. Zohar said that you were always guaranteed to have at least three people in the room for your talk: There was the speaker, the session chair, and the next speaker. Hence the three-person rule.

[11] https://rjlipton.wordpress.com/2014/11/12/three-in-the-room.

Zohar's favorite restaurant was the famous Sicilian Caffe Sport, in San Francisco; at least once a week, he would bring family, students, friends, and colleagues there for dinner. The staff would greet him like family, and he would regularly admonish the chef that the food did not have enough garlic—a blatant calumny.

Zohar's students all recount how much more than just superb academic guidance they received from him. Each has volumes of tales to tell of the sound advice, sage counsel, *joie de vivre*, and the vibrant example of both hard work and great play set by Zohar. (See Fig. 2.) And Zohar's devotion to his students is legendary. Just one tiny, emblematic example: When the Dershowitzes first arrived at Stanford, Zohar was out of town. Nitza collected Nachum and wife, Schulamith, from the San Francisco airport at 3 a.m., asserting contentedly that—unlike so many of their Bay Area acquaintances—they were still happily married. Zohar and Nitza proceeded to host the newlywed couple at their home until they found a place for themselves.

Above all, family and friends remember a person who enjoyed the fullness of life, was humble despite his many accomplishments, and was devoted to all who came into his orbit.

He will be missed sorely.

Bibliography

[1] Manna Z (1968) Termination of algorithms. PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA

[2] Manna Z (1969) Properties of programs and the first-order predicate calculus. J Assoc Comput Mach 16(2):244–255

[3] Manna Z (1969) The correctness of programs. J Comput Syst Sci 3(2):119–127

[4] Manna Z, McCarthy J (1969) Properties of programs and partial function logic. In: Meltzer B, Michie D (eds) Machine intelligence, vol 5. Edinburgh University Press, Edinburgh, pp 27–37

[5] Manna Z, Pnueli A (1969) Formalization of properties of recursively defined functions. In: Symposium on theory of computing (STOC). ACM, pp 201–210

[6] Manna Z, Ness S (1970) On the termination of Markov algorithms. In: Hawaii international conference on system sciences, Honolulu, HI, pp 789–792

[7] Manna Z (1970) Second-order mathematical theory of computation. Memo AIM-111. Stanford Artificial Intelligence Project, Stanford University

[8] Manna Z (1970) Termination of programs represented as interpreted graphs. In: Spring joint computing conference, vol 36. AFIPS, pp 83–89

[9] Manna Z (1970) Second-order mathematical theory of computation. In: Symposium on theory of computing (STOC). ACM, pp 158–168

[10] Manna Z (1970) The correctness of nondeterministic programs. Artif Intell 1(1/2):1–26

[11] Manna Z, Pnueli A (1970) Formalization of properties of functional programs. J. Assoc. Comput. Mach. 17(3):555–569

[12] Manna Z, Waldinger RJ (1970) Towards automatic program synthesis. Technical report CS-174 (AIM-127). Computer Science Dept., Stanford University

[13] Manna Z, Waldinger RJ (1970) Towards automatic program synthesis. Technical report 34. Artificial Intelligence Group, Stanford Research Institute

[14] Manna Z, Waldinger RJ (1970) On program synthesis and program verification. Technical note 52. AI Center, Stanford Research Institute

[15] Ashcroft EA, Manna Z (1971) The translation of "go to" programs to "while" programs. Technical report CS-188 (AIM-138). Computer Science Dept., Stanford University

[16] Manna Z (1971) Mathematical theory of partial correctness. Technical report CS-189 (AIM-139). Computer Science Dept., Stanford University

[17] Manna Z, Waldinger RJ (1971) Toward automatic program synthesis. Commun ACM 14(3):151–165

[18] Ashcroft EA, Manna Z (1971) Formalization of properties of parallel programs. In: Meltzer B, Michie D (eds) Machine intelligence, vol 6. Edinburgh University Press, Edinburgh, pp 17–41

[19] Chandra AK, Manna Z (1971) Program schemas with equality. Technical report CS-250 (AIM-158). Computer Science Dept., Stanford University

[20] Manna Z (1971) Mathematical theory of partial correctness. J Comput Syst Sci 5(3):239–253

[21] Manna Z (1971) Mathematical theory of partial correctness. In: Symposium on semantics of algorithmic languages, vol 188. Lecture notes in mathematics. Springer, pp 252–269

[22] Manna Z, Waldinger RJ (1971) Towards automatic program synthesis. In: Engeler E (ed) Symposium on semantics of algorithmic languages. Lecture notes in mathematics, vol 188. Springer, Berlin, pp 270–310

[23] Ashcroft EA, Manna Z, Pneuli A (1971) Decidable properties of monadic functional schemas. Technical report CS-217 (AIM-148). Computer Science Dept., Stanford University

[24] Manna Z, Ness S, Vuillemin J (1971) Inductive methods for proving properties of programs. Memo STAN-CS-71-243 (AIM-154). Dept. of Computer Science, Stanford University, Stanford, CA

[25] Cadiou J-M, Manna Z (1972) Recursive definitions of partial functions and their computations. In: Conference on proving assertions about programs. ACM, Las Cruces, NM, pp 58–65

[26] Manna Z, Ness S, Vuillemin J (1972) Inductive methods for proving properties of programs. In: Conference on proving assertions about programs. ACM, Las Cruces, NM, pp 27–50

[27] Manna Z, Vuillemin J (1972) Fixpoint approach to the theory of computation. Technical report CS-TR-71-272 (AIM-164). Computer Science Dept., Stanford University

[28] Chandra AK, Manna Z (1972) Program schemas with equality. In: Symposium on theory of computing (STOC). ACM, pp 52–64

[29] Manna Z (1972) Computation of recursive programs: theory vs. practice. In: Spring joint computing conference, vol 40. AFIPS, pp 219–223

[30] Manna Z, Vuillemin J (1972) Fixpoint approach to the theory of computation. In: International colloquium on automata,, languages and programming (ICALP), pp 273–291

[31] Manna Z, Vuillemin J (1972) Fixpoint approach to the theory of computation. Commun ACM 15(7):528–536

[32] Ashcroft EA, Manna Z (1972) The translation of "go to" programs to "while" programs. In: Proceedings of IFIP congress, vol 1. Foundations and systems. North-Holland, Amsterdam, pp 250–255

[33] Chandra AK, Manna Z (1973) On the power of programming features. Technical report CS-TR-73-333 (AIM-185). Dept. of Computer Science, Stanford University

[34] Katz SM, Manna Z (1973) A heuristic approach to program verification. Technical report CS-TR-73-341 (AIM-191). Dept. of Computer Science, Stanford University

[35] Ashcroft EA, Manna Z, Pnueli A (1973) Decidable properties of monadic functional schemas. J Assoc Comput Mach 20(3):489–499

[36] Manna Z, Pnueli A (1973) Axiomatic approach to total correctness of programs. Technical report CS-TR-73-382 (AIM-210). Dept. of Computer Science, Stanford University

[37] Katz S, Manna Z (1973) A heuristic approach to program verification. In: International joint conference on artificial intelligence (IJCAI). William Kaufmann, pp 500–512

[38] Manna Z, Stephen N, Vuillemin J (1973) Inductive methods for proving properties of programs. Commun ACM 16(8):491–502

[39] Manna Z (1974) Mathematical theory of computation. McGraw-Hill, New York

[40] Manna Z (1974) Knowledge and reasoning in program synthesis. In: Programming methodology, vol. 23. Lecture notes in computer science. Springer, pp 236–277

[41] Manna Z, Waldinger R (1974) Knowledge and reasoning in program synthesis. Report, Weizmann Institute of Science, Rehovot, Israel

[42] Manna Z, Pnueli A (1974) Axiomatic approach to total correctness of programs. Acta Inf 3(3):243–263

[43] Manna Z, Waldinger RJ (1974) Knowledge and reasoning in program synthesis. Technical note 98. AI Center, Stanford Research Institute

[44] Katz SM, Manna Z (1975) Towards automatic debugging of programs. In: International conference on reliable software, Los Angeles, CA, pp 143–155

[45] Ashcroft EA, Manna Z (1975) Translating program schemas to while-schemas. SIAM J Comput 4(2):125–146

[46] Katz S, Manna Z (1975) A closer look at termination. Acta Inf 5:333–352

[47] Manna Z, Shamir A (1975) The optimal fixedpoint of recursive programs. In: Symposium on theory of computing (STOC). ACM, pp 194–206

[48] Manna Z, Waldinger RJ (1975) Knowledge and reasoning in program synthesis. Artif Intell 6(2):175–208

[49] Manna Z, Waldinger RJ (1975) Knowledge and reasoning in program synthesis. In: International joint conference on artificial intelligence (IJCAI), pp 288–295

[50] Dershowitz N, Manna Z (1975) On automating structured programming. In: IRIA symposium on proving and improving programs. Arc-et-Senans, France, pp 167–193

[51] Dershowitz N, Manna Z (1975) On automating structured programming. Report, Dept. of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

[52] Manna Z, Shamir A (1975) A new approach to recursive programs. Technical report CS-TR-75-539 (AIM-276). Dept. of Computer Science, Stanford University

[53] Manna Z, Shamir A (1976) The theoretical aspects of the optimal fixedpoint. Technical report CS-TR- 76-542 (AIM-277). Dept. of Computer Science, Stanford University

[54] Katz S, Manna Z (1976) Logical analysis of programs. Commun ACM 19(4):188–206

[55] Manna Z, Shamir A (1976) The theoretical aspects of the optimal fixed point. SIAM J Comput 5(3):414–426

[56] Manna Z, Waldinger R (1976) Is "sometime" sometimes better than "always"? Intermittent assertions in proving program correctness. Memo AIM-281. Stanford Artificial Intelligence Laboratory, Stanford University

[57] Manna Z, Waldinger R (1976) Is "sometime" sometimes better than "always"? intermittent assertions in proving program correctness. Technical note 132. AI Center, Stanford Research Institute

[58] Manna Z, Waldinger RJ (1976) Is "sometime" sometimes better than "always"? Intermittent assertions in proving program correctness. In: International conference on software engineering (ICSE). IEEE, pp 32–39

[59] Chandra AK, Manna Z (1976) On the power of programming features. Comput Lang 1(3):219–232

[60] Dershowitz N, Manna Z (1976) The evolution of programs: a system for automatic program modification. Memo AIM-294 (CS-TR-76-586). Stanford Artificial Intelligence Laboratory, Stanford University, Stanford, CA

[61] Manna Z, Waldinger R (1977) Is "sometime" sometimes better than "always"? Intermittent assertions in proving program correctness. Technical report CS-TR-76-558 (AIM-281.1). Dept. of Computer Science, Stanford University

[62] Manna Z, Shamir A (1977) The convergence of functions to fixedpoints of recursive definitions. Technical report CS-TR-77-614 (AIM-299). Dept. of Computer Science, Stanford University

[63] Dershowitz N, Manna Z (1977) The evolution of programs: automatic program modification. IEEE Trans Softw Eng 3(6):377–385

[64] Dershowitz N, Manna Z (1977) The evolution of programs: a system for automatic program modification. In: International conference on principles of programming languages (POPL). ACM, pp 144–154

[65] Manna Z, Shamir A (1977) The optimal approach to recursive programs. Commun ACM 20(11):824–831

[66] Manna Z, Waldinger R (1977) Studies in automatic programming logic. With contributions by Sh. Katz and K. Levitt. Elsevier, New York

[67] Manna Z, Waldinger R (1977) The automatic synthesis of recursive programs. SIGART Newsl 64:29–36

[68] Manna Z, Waldinger RJ (1977) The automatic synthesis of systems of recursive programs. In: International joint conference on artificial intelligence (IJCAI). William Kaufmann, pp 405–411

[69] McCarthy J, Binford TO, Green CC, Luckham DC, Manna Z, Winograd TA, Earnest LD (1977) Recent research in computer science. Technical report CS-TR-77-624. Dept. of Computer Science, Stanford University

[70] Manna Z, Waldinger R (1977) The logic of computer programming. Technical report STAN-CS- 77-611 (AIM-298). Dept. of Computer Science, Stanford University, Stanford, CA

[71] Waldinger R, Manna Z (1977) The logic of computer programming. Technical note 154. AI Center, SRI International

[72] Dershowitz N, Manna Z (1977) Inference rules for program annotation. Memo AIM-303 (CS-TR-77-631). Stanford Artificial Intelligence Laboratory, Stanford University, Stanford, CA

[73] Manna Z, Waldinger R (1977) Synthesis: dreams ⇒ programs. Technical report STAN-CS-77-630 (AIM-302). Dept. of Computer Science, Stanford University, Stanford, CA

[74] Manna Z, Waldinger R (1977) Synthesis: dreams ⇒ programs. Technical note 156. AI Center, SRI International

[75] Manna Z, Waldinger R (1978) Structured programming with recursion. Technical report STAN-CS- 77-640 (AIM-307). Dept. of Computer Science, Stanford University, Stanford, CA

[76] Dershowitz N, Manna Z (1978) Proving termination with multiset orderings. Memo AIM-310 (CS-TR-78-651). Stanford Artificial Intelligence Laboratory, Stanford University, Stanford, CA

[77] Dershowitz N, Manna Z (1978) Inference rules for program annotation. In: International conference on software engineering (ICSE). IEEE, pp 158–167

[78] Manna Z, Shamir A (1978) The convergence of functions to fixedpoints of recursive definitions. Theor Comput Sci 6(2):109–141

[79] Manna Z, Waldinger R (1978) DEDALUS - The DEDuctive ALgorithm Ur-Synthesizer. In: International workshop managing requirements knowledge, pp 683–689

[80] Manna Z, Waldinger RJ (1978) Is "sometime" sometimes better than "always"? intermittent assertions in proving program correctness. Commun ACM 21(2):159–172

[81] Manna Z, Waldinger RJ (1978) The logic of computer programming. IEEE Trans Softw Eng 4(3):199–229

[82] Manna Z, Waldinger R (1978) The synthesis of structure changing programs. In: International conference on software engineering (ICSE). IEEE, pp 175–187

[83] Manna Z (1978) Six lectures on the logic of computer programming. Technical report STAN-CS-78-688 (AIM-318). Dept. of Computer Science, Stanford University, Stanford, CA

[84] Manna Z, Waldinger RJ (1978) A deductive approach to program synthesis. Technical report CS-TR-78-690 (AIM-320). Dept. of Computer Science, Stanford University

[85] Manna Z, Waldinger R (1978) A deductive approach to program synthesis. Technical note 177. AI Center, SRI International

[86] Ashcroft EA, Manna Z (1979) The translation of "go to" programs to "while" programs. In: Yourdon EN (ed) Classics in software engineering. Yourdon Press, Upper Saddle River, pp 49–61

[87] Manna Z, Waldinger R (1979) A deductive approach to program synthesis. In: International joint conference artificial intelligence (IJCAI). William Kaufmann, pp 542–551

[88] Manna Z, Waldinger RJ (1979) Synthesis: dreams ⇒ programs. IEEE Trans Softw Eng 5(4):294–328

[89] Dershowitz N, Manna Z (1979) Proving termination with multiset orderings. In: International colloquium on automata, languages and programming (ICALP), vol 71. Lecture notes in computer science. Springer, pp 188–202

[90] Manna Z, Pnueli A (1979) The modal logic of programs. In: International colloquium on automata, languages and programming (ICALP), vol 71. Lecture notes in computer science. Springer, pp 385–409

[91] Dershowitz N, Manna Z (1979) Proving termination with multiset orderings. Commun ACM 22(8):465–476

[92] Manna Z, Pnueli A (1979) The modal logic of programs. Technical report STAN-CS-78-751 (AIM-330). Dept. of Computer Science, Stanford University, Stanford, CA

[93] Manna Z (1980) Synthesized structured programming. Final technical report RADC-TR-79-326. With help by N. Dershowitz. Rome Air Development Center, Griffiss AFB, NY

[94] Manna Z, Pnueli A (1980) Synchronous schemes and their decision problems. In: International conference on principles of programming languages (POPL). ACM, pp 62–67

[95] Manna Z (1980) The logic of computer programming. In: Arden BW (ed) What can be automated: the computer science and engineering research study (COSERS). MIT Press, Cambridge, pp 235–260

[96] McCarthy J, Binford TO, Luckham DC, Manna Z, Weyhrauch RW, Earnest L (1980). Final report: basic research in artificial intelligence and foundations of programming. Technical report CS-TR-1980-808 (AIM-337). Dept. of Computer Science, Stanford University

[97] Manna Z (1980) Lectures on the logic of computer programming, vol 31. CBMS-NSF regional conference series in applied mathematics. With contributions by N. Dershowitz and R. Waldinger. SIAM

[98] Manna Z, Waldinger RJ (1980) A deductive approach to program synthesis. ACM Trans Program Lang Syst 2(1):90–121

[99] Manna Z, Waldinger R (1980) Problematic features of programming languages: a situational-calculus approach. Technical report STAN-CS-80-779. Dept. of Computer Science, Stanford University, Stanford, CA

[100] Manna Z (1980) Logics of programs. In: Lavington SH (ed) Information processing 80. IFIP, Amsterdam, pp 41–51

[101] Manna Z, Waldinger R (1980) Problematic features of programming languages: situational-calculus, approach part I: assignment statements. Technical note 226. AI Center, SRI International

[102] Ben-Ari M, Manna Z, Pnueli A (1981) The temporal logic of branching time. In: International conference on principles of programming languages (POPL). ACM, pp 164–176

[103] Dershowitz N, Manna Z (1981) Inference rules for program annotation. IEEE Trans Softw Eng 7(2):207–222

[104] Manna Z, Pnueli A (1981) Temporal verification of concurrent programs, part I: the temporal framework for concurrent programs. Technical report STAN-CS-81-836. Dept. of Computer Science, Stanford University, Stanford, CA

[105] Manna Z, Waldinger R (1981) Deductive synthesis of the unification algorithm. Technical report STAN-CS-81-855. Dept. of Computer Science, Stanford University, Stanford, CA

[106] Manna Z, Waldinger RJ (1981) Problematic features of programming languages: a situational-calculus approach. Acta Inf 16(4):371–426

[107] Manna Z, Waldinger RJ (1981) Deductive synthesis of the unification algorithm. Sci Comput Program 1(1–2):5–48

[108] Manna Z, Waldinger R (1981) Deductive synthesis of the unification algorithm. Technical note 246. AI Center, SRI International

[109] Manna Z (1981) Verification of sequential programs: temporal axiomatization. Technical report STAN-CS-81-877. Dept. of Computer Science, Stanford University, Stanford, CA

[110] Manna Z, Pnueli A (1981) Verification of concurrent programs, part II: temporal proof principles. Technical report STAN-CS-81-843. Air Force Office of Scientific Research Report AFOSR-TR-82-0495. Dept. of Computer Science, Stanford University, Stanford, CA

[111] Manna Z, Wolper P (1981) Synthesis of communicating processes from temporal logic specifications. Technical report STAN-CS-81-872. Dept. of Computer Science, Stanford University, Stanford, CA

[112] Manna Z, Waldinger R (1982) Special relations in program-synthetic deduction. Technical report STAN-CS-82-902. Dept. of Computer Science, Stanford University, Stanford, CA

[113] Manna Z, Pnueli A (1982) Verification of concurrent programs: proving eventualities by well-founded ranking. Technical report STAN-CS-82-915. Dept. of Computer Science, Stanford University, Stanford, CA

[114] Manna Z (1982) Verification of sequential programs: temporal axiomatization. In: Broy M, Schmidt G (eds) Theoretical foundations of programming methodology: lecture notes of an international summer school, vol 91. NATO advanced science institutes series. D. Reidel, Dordrecht, pp 53–102

[115] Manna Z, Pnueli A (1982) Verification of concurrent programs: temporal proof principles. In: Kozen D (ed) Proceedings of the of workshop on logics of programs, vol 131. Lecture notes in computer science. Springer, Berlin, pp 200–252

[116] Manna Z, Pnueli A (1982) Verification of concurrent programs: the temporal framework. In: Boyer RS, Moore JS (eds) The correctness problem in computer science. International lecture series in computer science. Academic Press, London, pp 215–273

[117] Manna Z, Wolper P (1982) Synthesis of communicating processes from temporal logic specifications. In: Kozen D (ed) Proceedings of the of workshop on logics of programs, vol 131. Lecture notes in computer science. Springer, Berlin, pp 253–281

[118] Manna Z, Pnueli A (1982) How to cook a temporal proof system for your pet language. Technical report STAN-CS-82-954. Dept. of Computer Science, Stanford University, Stanford, CA

[119] Manna Z, Pnueli A (1982) Verification of concurrent programs: a temporal proof system. Technical report STAN-CS-83-967. Dept. of Computer Science, Stanford University, Stanford, CA

[120] Manna Z, Pnueli A (1983) How to cook a temporal proof system for your pet language. In: International conference on principles of programming languages (POPL). ACM, pp 141–154

[121] Manna Z, Pnueli A (1983) Proving precedence properties: the temporal way. Technical report, Dept. of Computer Science, Stanford University, Stanford, CA

[122] Halpern JY, Manna Z, Moszkowski BC (1983) A hardware semantics based on temporal intervals. In: International colloquium on automata, languages and programming (ICALP), vol 154. Lecture notes in computer science. Springer, pp 278–291

[123] Manna Z, Pnueli A (1983) Verification of concurrent programs: a temporal proof system. In: Bakker JW, Van Leeuwen J (eds) Foundations of computer science: distributed systems: part 1. Algorithms and complexity, vol 159. Mathematical centre tracts. Mathematisch Centrum, Amsterdam, pp 163–255

[124] Manna Z, Waldinger R (1983) Deductive synthesis of the unification algorithm. In: Biermann AW, Guiho G (eds) Computer program synthesis methodologies, vol 95. NATO advanced study institutes series (series C: mathematical and physical sciences). Springer, Dordrecht, pp 251–307

[125] Manna Z, Waldinger RJ (1983) The deductive foundations of computer programming. Addison-Wesley, Reading

[126] Manna Z, Pnueli A (1983) Proving precedence properties: the temporal way. In: International colloquium on automata, languages and programming (ICALP), vol 154. Lecture notes in computer science. Springer, pp 491–512

[127] Ben-Ari M, Pnueli A, Manna Z (1983) The temporal logic of branching time. Acta Inf 20(3):207–0226

[128] Manna Z, Waldinger R (1984) Special relations in program-synthetic deduction. In: Lassez J-L, Crossley J (eds) Proceedings of logic and computation conference, Melbourne, Australia

[129] Manna Z, Wolper P (1984) Synthesis of communicating processes from temporal logic specifications. ACM Trans Program Lang Syst 6(1):68–93

[130] Manna Z, Pnueli A (1984) Proving precedence properties: the temporal way. Report, Weizmann Institute of Science, Rehovot, Israel

[131] Manna Z, Waldinger R (1984) Deductive programming. In: Jerusalem conference on information technology (JCIT): next decade in information technology. IEEE

[132] Manna Z (1984) Deductive programming. Interim report AFOSR-TR-84-0297. Air Force Office of Scientific Research

[133] Manna Z, Waldinger R (1984) A deductive approach to program synthesis. In: Biermann AW, Guiho G, Kodratoff Y (eds) Automatic program construction techniques. Macmillan, New York, pp 33–68

[134] Manna Z, Waldinger R (1984) Problematic features of programming languages. In: Biermann AW, Guiho G, Kodratoff Y (eds) Automatic program construction techniques. Macmillan, New York, pp 109–121

[135] Moszkowski B, Manna Z (1984) Reasoning in interval temporal logic. In: Proceedings of workshop logic of programs, vol 164. Lecture notes in computer science. Springer, pp 371–382

[136] Malachi Y, Manna Z, Waldinger RJ (1984) TABLOG: the deductive-tableau programming language. In: LISP and functional programming, pp 323–330

[137] Waldinger R, Malachi Y, Manna Z (1984) TABLOG: the deductive-tableau programming language. Technical note 328. AI Center, SRI International

[138] Manna Z, Pnueli A (1984) Adequate proof principles for invariance and liveness properties of concurrent programs. Sci Comput Program 4(3):257–289

[139] Malachi Y, Manna Z, Waldinger R (1985) TABLOG: a new approach to logic programming. Technical report STAN-CS-86-1110. Dept. of Computer Science, Stanford University, Stanford, CA

[140] Manna Z, Waldinger R (1985) The origin of the binary-search paradigm. Technical report STAN-CS-85-1044. Dept. of Computer Science, Stanford University, Stanford, CA

[141] Abadi M, Manna Z (1985) Nonclausal temporal deduction. In: Proceedings of the logic of programs conference, vol 193. Lecture notes in computer science. Springer, pp 1–15

[142] Manna Z, Waldinger R (1985) Special relations in automated deduction. Technical note 355. AI Center, SRI International

[143] Manna Z, Waldinger RJ (1985) Logical basis for computer programming. Vol. 1. Deductive reasoning. Addison-Wesley, Reading

[144] Manna Z, Waldinger RJ (July 1985) Special relations in automated deduction. In: International colloquium on automata, languages and programming (ICALP), vol 194. Lecture notes in computer science. Springer, pp 413–423

[145] Manna Z, Waldinger RJ (1985) The origin of the binary-search paradigm. In: International joint conference on artificial intelligence (IJCAI). Morgan Kaufmann, pp 222–224

[146] Manna Z, Waldinger RJ (1985) Deduction with relation matching. In: Conference on foundations of software technology and theoretical computer science (FSTTCS), vol 206. Lecture notes in computer science. Springer, pp 212–224

[147] Malachi Y, Manna Z, Waldinger RJ (1986) TABLOG: functional and relational programming in one framework. IEEE Softw 3(1):75–76

[148] Manna Z, Waldinger RJ (1986) Special relations in automated deduction. J Assoc Comput Mach 33(1):1–59

[149] Abadi M, Manna Z (1986) Modal theorem proving. Technical report STAN-CS-86-1110. Dept. of Computer Science, Stanford University, Stanford, CA

[150] Abadi M, Manna Z (1986) A timely resolution. In: International symposium on logic in computer science (LICS). IEEE, pp 176–186

[151] Jonsson B, Manna Z, Waldinger RJ (1986) Towards deductive synthesis of dataflow networks. In: International symposium on logic in computer science (LICS). IEEE, pp 26–37

[152] Malachi Y, Manna Z, Waldinger RJ (1986) TABLOG: a new approach to logic programming. In: DeGroot D, Lindstrom G (eds) Logic programming: functions, relations, and equations, pp 365–394

[153] Manna Z, Waldinger R (1986) A deductive approach to program synthesis. In: Rich C, Waters RC (eds) Artificial intelligence and software engineering. Morgan Kaufmann, Los Altos, pp 3–34

[154] Abadi M, Manna Z (1986) Modal theorem proving. In: International conference on automated deduction (CADE), vol 230. Lecture notes in computer science. Springer, pp 172–189

[155] Manna Z, Waldinger RJ (1986) How to clear a block: a theory of plans. In: Georgeff MP, Lansky AL (eds) Proceedings of workshop reasoning about actions and plans, Timberline, OR, pp 11–45

[156] Manna Z, Waldinger RJ (1986) How to clear a block: plan formation in situational logic. In: International conference on automated deduction (CADE), vol 230. Lecture notes in computer science. Springer, pp 622–640

[157] Waldinger R, Manna Z (1986) The origin of the binary-search paradigm. Technical note 351. AI Center, SRI International

[158] Manna Z, Waldinger R (1986) How to clear a block: a theory of plans. Technical note 397. AI Center, SRI International

[159] Manna Z, Pnueli A (1987) Specification and verification of concurrent programs by ∀-automata. In: International conference on principles of programming languages (POPL). ACM, pp 1–12

[160] Manna Z, Pnueli A (1987) A hierarchy of temporal properties (abstract). In: Symposium on principles of distributed computing (PODC). ACM, p 205

[161] Manna Z, Waldinger RJ (1987) The deductive synthesis of imperative LISP programs. In: AAAI conference on artificial intelligence. Morgan Kaufmann, pp 155–160

[162] Manna Z, Waldinger RJ (1987) The origin of a binary-search paradigm. Sci Comput Program 9(1):37–83

[163] Abadi M, Manna Z (1987) Temporal logic programming. In: Symposium on logic programming (SLP). IEEE, pp 4–16

[164] Manna Z, Waldinger RJ (1987) How to clear a block: a theory of plans. J Autom Reason 3(4):343–377

[165] Manna Z, Pnueli A (1988) The anchored version of the temporal framework. In: de Bakker JW, de Roever W-P, Rozenberg G (eds) Linear time, branching time and partial order in logics and models for concurrency (REX 1988 school/workshop), vol 354. Lecture notes in computer science. Springer, Berlin, pp 201–284

[166] Manna Z (1988) Deductive programming. Final technical report Dept. of the Navy Contract N00039-84-C-0211 (task 3). Defense Advanced Research Projects Agency (DoD), Arlington, VA

[167] Manna Z, Pnueli A (1989) Specification and verification of concurrent programs by ∀-automata. In: Proceedings of temporal logic in specification, vol 398. Lecture notes in computer science. Springer, pp 124–164

[168] Manna Z, Pnueli A (1989) Completing the temporal picture. In: International colloquium on automata, languages and programming (ICALP), vol 372. Lecture notes in computer science. Springer, pp 534–558

[169] Abadi M, Manna Z (1989) Temporal logic programming. J Symb Comput 8(3):277–295

[170] Abadi M, Manna Z (1990) Nonclausal deduction in first-order temporal logic. J Assoc Comput Mach 37(2):279–317

[171] Manna Z, Pnueli A (1990) An exercise in the verification of multi-process programs. In: Feijen WHJ, van Gasteren AJM, Gries D, Misra J (eds) Beauty is our business. Springer, Berlin, pp 289–301

[172] Manna Z, Waldinger RJ (1990) Logical basis for computer programming, vol 2. Deductive systems. Addison-Wesley, Reading

[173] Manna Z, Pnueli A (1990) Tools and rules for the practicing verifier. Technical report CS-TR-90-1321. Dept. of Computer Science, Stanford University

[174] Manna Z, Pnueli A (1990) A hierarchy of temporal properties. In: Symposium on principles of distributed computing (PODC). ACM, pp 377–410

[175] Henzinger TA, Manna Z, Pnueli A (1990) An interleaving model for real time. Technical report CS-TR-90-1329. Dept. of Computer Science, Stanford University

[176] Henzinger TA, Manna Z, Pnueli A (1990) An interleaving model for real-time. In: Jerusalem conference on information technology (JCIT). IEEE, pp 717–730

[177] Manna Z, Pnueli A (1990) A temporal proof methodology for reactive systems. In: Jerusalem conference on information technology (JCIT). IEEE, pp 757–773

[178] Henzinger TA, Manna Z, Pnueli A (1991) Temporal proof methodologies for real-time systems. In: International conference on principles of programming languages (POPL). ACM, pp 353–366

[179] Manna Z (1991) Deductive programming synthesis. Final technical report DARPA Order No. 6177. Defense Advanced Research Projects Agency (DoD), Arlington, VA

[180] Manna Z, Pnueli A (1991) Completing the temporal picture. Theor Comput Sci. 83(1):97–130

[181] Manna Z, Pnueli A (1991) Tools and rules for the practicing verifier. In: Rashid RF (ed) CMU computer science: a 25th anniversary commemorative. ACM Press, New York, pp 125–159

[182] Manna Z, Stickel M, Waldinger R (1991) Monotonicity properties in automated deduction. In: Lifschitz V (ed) Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy. Academic Press, New York, pp 261–280

[183] Manna Z, Waldinger R (1991) Fundamentals of deductive program synthesis. In: Bauer FL (ed) Logic, algebra, and computation, vol 79. NATO advanced science institutes series. Subseries F: computer and system sciences. Springer, Berlin, pp 41–107

[184] Henzinger TA, Manna Z, Pnueli A (1991) Temporal proof methodologies for real-time systems. Technical report CS-TR-91-1383. Dept. of Computer Science, Stanford University

[185] Manna Z, Pnueli A (1991) On the faithfulness of formal models. In: Mathematical foundations of computer science (MFCS), vol 520. Lecture notes in computer science. Springer, pp 28–42

[186] Manna Z, Waldinger R (1992) Fundamentals of deductive program synthesis. Technical report STAN-CS-92-1404. Dept. of Computer Science, Stanford University, Stanford, CA

[187] Manna Z, Pnueli A (1992) Temporal specification and verification of reactive modules. Report, Weizmann Institute of Science, Rehovot, Israel

[188] Henzinger TA, Manna Z, Pnueli A (1992) Timed transition systems. In: Real-time: theory in practice (REX workshop), vol 600. Lecture notes in computer science. Springer, pp 226–251

[189] Maler O, Manna Z, Pnueli A (1992) From timed to hybrid systems. In: Real-time: theory in practice (REX workshop), vol 600. Lecture notes in computer science. Springer, pp 447–484

[190] Manna Z, Pnueli A (1992) The temporal logic of reactive and concurrent systems: specification. Springer, New York

[191] Manna Z, Waldinger RJ (June 1992) The special-relation rules are incomplete. In: International conference on automated deduction (CADE), vol 607. Lecture notes in computer science. Springer, pp 492–506

[192] Chang EY, Manna Z, Pnueli A (1992) Characterization of temporal property classes. In: International colloquium on automata, languages and programming (ICALP), vol 623. Lecture notes in computer science. Springer, pp 474–486

[193] Henzinger TA, Manna Z, Pnueli A (1992) What good are digital clocks? In: International colloquium on automata, languages and programming (ICALP), vol 623. Lecture notes in computer science. Springer, pp 545–558

[194] Manna Z, Waldinger RJ (1992) Fundamentals of deductive program synthesis. IEEE Trans Softw Eng 18(8):674–704

[195] Waldinger R, Manna Z (1992) Fundamentals of deductive program synthesis. Technical note 503. AI Center, SRI International

[196] Manna Z, Pnueli A (1992) Time for concurrency. In: Future tendencies in computer science, control and applied mathematics. International conference of the occasion of the 25th anniversary of INRIA, vol 653. Lecture notes in computer science. Springer, pp 129–153

[197] Manna Z, Pnueli A (1993) A temporal proof methodology for reactive systems. Technical report STAN-CS-92-1463. Extended version. Dept. of Computer Science, Stanford University, Stanford, CA

[198] Chang E, Manna E, Pnueli A (1993) The safety-progress classification. In: Bauer FL, Brauer W, Schwichtenberg H (eds) Logic and algebra of specifications, vol 94. NATO advanced science institutes series. Springer, Berlin, pp 143–202

[199] Henzinger TA, Manna Z, Pnueli A (1993) Towards refining temporal specifications into hybrid systems. In: International conference on hybrid systems, vol 736. Lecture notes in computer science. Springer, pp 60–76

[200] Kesten Y, Manna Y, McGuire H, Pnueli A (1993) A decision algorithm for full prepositional temporal logic. In: International conference on computer-aided verification (CAV), vol 697. Lecture notes in computer science. Springer, pp 97–109

[201] Manna Z, Pnueli A (1993) Models for reactivity. Acta Inf 30(7):609–678

[202] Manna Z, Pnueli A (1993) Verifying hybrid systems. In: International conference on hybrid systems, vol 736. Lecture notes in computer science. Springer, pp 4–35

[203] Manna Z, Pnueli A (1993) A temporal proof methodology for reactive systems. In: Broy M (ed) Program design calculi, vol 118. NATO advanced science institutes series. Series F: computer and system sciences. Springer, Berlin, pp 287–323

[204] Manna Z, Pnueli A (1994) Temporal verification diagrams. In: International symposium on theoretical aspects of computer software (TACS), vol 789. Lecture notes in computer science. Springer, pp 726–765

[205] Anuchitanukul A, Manna Z (1994) Realizability and synthesis of reactive modules. In: International conference on computer-aided verification (CAV), vol 818. Lecture notes in computer science. Springer, pp 156–168

[206] Chang EY, Manna Z, Pnueli A (1994) Compositional verification of real-time systems. In: International symposium on logic in computer science (LICS). IEEE, pp 458–465

[207] Henzinger TA, Manna Z, Pnueli A (1994) Temporal proof methodologies for timed transition systems. Inf Comput 112(2):273–337

[208] Kesten Y, Manna Z, Pnueli A (1994) Temporal verification of simulation and refinement. In: A decade of concurrency (REX school/symposium), vol 803. Lecture notes in computer science. Springer, pp 273–346

[209] Manna Z (1994) Beyond model checking. In: International conference on computer-aided verification (CAV), vol 818. Lecture notes in computer science. Springer, pp 220–221

[210] Manna Z, Anuchitanukul A, Bjłrner N, Browne A, Chang E, Colón M, de Alfaro L, Devarajan H, Sipma H, Uribe T (1994) STeP: the Stanford temporal prover. Technical report CS-TR-94-1518. Dept. of Computer Science, Stanford University

[211] McGuire H, Manna Z, Waldinger RJ (1994) Annotation-based deduction in temporal logic. In: International conference on temporal logic (ICTL), vol 827. Lecture notes in computer science. Springer, pp 430–444

[212] de Alfaro L, Manna Z (1994) Continuous verification by discrete reasoning. Technical report CS-TR- 94-1524. Dept. of Computer Science, Stanford University

[213] Anuchitanukul A, Manna Z (1994) Differential BDDs. Technical report CS-TR-94-1525. Dept. of Computer Science, Stanford University

[214] Kapur A, Henzinger TA, Manna Z, Pnueli A (1994) Proving safety properties of hybrid systems. In: Conference on formal techniques in real-time and fault tolerant system (FTRTFT), vol 863. Lecture notes in computer science. Springer, pp 431–454

[215] Sipma H, Manna Z (1994) Specification and verification of controlled systems. In: Conference on formal techniques in real-time and fault tolerant system (FTRTFT), vol 863. Lecture notes in computer science. Springer, pp 641–659

[216] Manna Z, Bjørner N, Browne IA, Chang EY, Colón M, de Alfaro L, Devarajan H, Kapur A, Lee J, Sipma H, Uribe TE (1995) STeP: the Stanford temporal prover. In: International joint conference on theory and practice of software development (TAPSOFT), vol 915. Lecture notes in computer science. Springer, pp 793–794

[217] Anuchitanukul A, Manna Z, Uribe TE (1995) Differential BDDs. In: Computer science today, vol 1000. Lecture notes in computer science. Springer, pp 218–233

[218] Manna Z, Paltrinieri M, Waldinger RJ (1995) A temporal plan theory. In: World conference on fundamentals of artificial intelligence (WOCFAI), pp 253–261

[219] Manna Z, Pnueli A (1995) Temporal verification of reactive systems: safety. Springer, New York

[220] Manna Z, Pnueli A (1995) Verification of parameterized programs. In: Börger E (ed) Specification and validation methods. Oxford University Press, Oxford, pp 167–230

[221] de Alfaro L, Manna Z (1995) Verification in continuous time by discrete reasoning. In: International conference on algebraic methodology and software technology (AMAST), vol 936. Lecture notes in computer science. Springer, pp 292–306

[222] Manna Z, Sipma HB (1995) A deductive approach towards controller synthesis. In: International symposium on intelligent control. IEEE, pp 35–41

[223] Bjørner N, Browne IA, Manna Z (1995) Automatic generation of invariants and intermediate assertions. In: International conference on principles and practice of constraint programming, vol 976. Lecture notes in computer science. Springer, pp 589–623

[224] Bjørner N, Browne A, Chang E, Colón M, Kapur A, Manna Z, Sipma HB, Uribe TE (1995) STeP: the Stanford temporal prover (educational release) user's manual. Technical report CS-TR-95-1562. Dept. of Computer Science, Stanford University

[225] Browne IA, Manna Z, Sipma H (1995) Generalized temporal verification diagrams. In: Conference on foundations of software technology and theoretical computer science (FSTTCS), vol 1026. Lecture notes in computer science. Springer, pp 484–498

[226] Manna Z, Pnueli A (1996) Clocked transition systems. Technical report CS-TR-96-1566. Dept. of Computer Science, Stanford University

[227] Kesten Y, Manna Z, Pnueli A (1996) Verifying clocked transition systems. In: Hybrid systems, vol 1066. Lecture notes in computer science. Springer, pp 13–40

[228] Manna Z, Pnueli A (1996) Clocked transition systems. In: Logic and software engineering: an international workshop in honor Chih-Sung Tang. World Scientific, Beijing, pp 3–42

[229] de Alfaro L, Manna Z (1996) Temporal verification by diagram transformations. In: International conference on computer-aided verification (CAV), vol 1102. Lecture notes in computer science. Springer, pp 288–299

[230] Bjørner N, Browne IA, Chang EY, Colón M, Kapur A, Manna Z, Sipma H, Uribe TE (1996) STeP: deductive-algorithmic verification of reactive and real-time systems. In: International conference on computer-aided verification (CAV), vol 1102. Lecture notes in computer science. Springer, pp 415–418

[231] Browne A, de Alfaro L, Manna Z, Sipma HB, Uribe TE (1996) Diagram-based formalisms for the verification of reactive systems. In: CADE-13 workshop on visual reasoning, New Brunswick, NJ

[232] Sipma H, Uribe TE, Manna Z (1996) Deductive model checking. In: International conference on computer-aided verification (CAV), vol 1102. Lecture notes in computer science. Springer, pp 208–219

[233] Browne IA, Manna Z, Sipma H (1996) Hierarchical verification using verification diagrams. In: Asian computing science conference (ASIAN), vol 1179. Lecture notes in computer science. Springer, pp 276–286

[234] de Alfaro L, Kapur A, Manna Z (1997) Hybrid diagrams: a deductive-algorithmic approach to hybrid system verification. In: International symposium on theoretical aspects of computer science (STACS), vol 1200. Lecture notes in computer science. Springer, pp 153–164

[235] Bjørner N, Browne IA, Manna Z (1997) Automatic generation of invariants and intermediate assertions. Theor Comput Sci 173(1):49–87

[236] de Alfaro L, Manna Z, Sipma HB, Uribe TE (1997) Visual verification of reactive systems. In: Tools and algorithms for the construction and analysis of systems (TACAS), vol 1217. Lecture notes in computer science. Springer, pp 334–350

[237] Bjørner N, Manna Z, Sipma H, Uribe TE (1997) Deductive verification of realtime systems using STeP. In: International AMAST workshop real-time systems and concurrent and distributed software (ARTS), vol 1231. Lecture notes in computer science. Springer, pp 22–43

[238] Manna Z (1997) Verification of progress properties. In: APPIA-GULP-PRODE joint conference on declarative programming (AGP), pp 345–346

[239] Bjørner N, Lerner U, Manna Z (1997) Deductive verification of parameterized fault-tolerant systems: a case study. In: International conference on temporal logic. Kluwer

[240] Manna Z, Browne IA, Sipma H, Uribe TE (1998) Visual abstractions for temporal verification. In: International conference on algebraic methodology and software technology (AMAST), vol 1548. Lecture notes in computer science. Springer, pp 28–41

[241] Manna Z, Sipma H (1998) Deductive verification of hybrid systems using STeP. In: Hybrid systems: computation and control, vol 1386. Lecture notes in computer science. Springer, pp 305–318

[242] Finkbeiner B, Manna Z, Sipma H (1998) Deductive verification of modular systems. In: Compositionality: the significant difference (COMPOS'98), vol 1536. Lecture notes in computer science. Springer, pp 239–275

[243] Kesten Y, Manna Z, Pnueli A (1998) Verification of clocked and hybrid systems. In: European educational forum: lectures on embedded systems, vol 1494. Lecture notes in computer science. Springer, pp 4–73

[244] Manna Z, Colón M, Finkbeiner B, Sipma H, Uribe TE (1998) Abstraction and modular verification of infinite-state reactive systems. In: Requirements targeting software and systems engineering (RTSE), vol 1526. Lecture notes in computer science. Springer, pp 273–292

[245] de Alfaro L, Manna Z, Sipma H (1998) Decomposing, transforming and composing diagrams: the joys of modular verification. Technical report CS-TR-98-1614. Dept. of Computer Science, Stanford University

[246] Manna Z (1999) Visual verification of temporal properties. In: International symposium on temporal representation and reasoning (TIME). IEEE, p 6

[247] Manna Z (1999) STeP: a tool for the development of provably correct reactive and real-time systems. Final technical report ARO 34456.8-MA. U.S. Army Research Office

[248] Manna Z, Bjørner N, Browne IA, Colon M, Finkbeiner B, Pichora M, Sipma HB, Uribe T (1999) An update on STeP: deductive-algorithmic verification of reactive systems. In: Tool support for system specification, development and verification, advances in computing science. Springer, pp 174–188

[249] Manna Z, Sipma H (1999) Verification of parameterized systems by dynamic induction on diagrams. In: International conference on computer-aided verification (CAV), vol 1633. Lecture notes in computer science. Springer, pp 25–41

[250] Sipma H, Uribe TE, Manna Z (1999) Deductive model checking. Form Methods Syst Des 15(1):49–74

[251] Bjørner N, Browne IA, Colón M, Finkbeiner B, Manna Z, Sipma H, Uribe TE (2000) Verifying temporal properties of reactive systems: a STeP tutorial. Form Methods Syst Des 16(3):227–270

[252] Browne IA, Finkbeiner B, Manna Z, Sipma H (2000) The cash-point service: a verification case study using STeP. Form Asp Comput 12(4):218–219

[253] Kesten Y, Manna Z, Pnueli A (2000) Verification of clocked and hybrid systems. Acta Inf 36(11):837–912

[254] Manna Z, Siprna HB (2000) Verification diagrams: logic + automata. In: Proceedings of workshop modeling software system structures in fastly moving scenario. Santa Margherita Ligure, Italy, pp 226–237

[255] Manna Z, Sipma H (2000) Alternating the temporal picture for safety. In: International Colloquium on automata, languages and programming (ICALP), vol 1853. Lecture notes in computer science. Springer, pp 429–450

[256] Bjørner N, Manna Z, Sipma H, Uribe TE (2001) Deductive verification of real-time systems using STeP. Theor Comput Sci 253(1):27–60

[257] Sastry SS, Nerode A, Manna Z (2002) An integrated approach to intelligent systems. Final report 35873.172-MA-MUR. U.S. Army Research Office

[258] Manna Z, Zarba CG (2003) Combining decision procedures. In: 10th anniversary colloquium of UNU/IIST, vol 2757. Lecture notes in computer science. Springer, pp 381–422

[259] Sankaranarayanan S, Sipma H, Manna Z (2003) Petri net analysis using invariant generation. In: Verification: theory and practice, vol 2772. Lecture notes in computer science. Springer, pp 682–701

[260] Zarba CG, Manna Z, Sipma HB (2003) Combining theories sharing dense orders. In: TABLEA UX2003: position papers and tutorials. Technical report RT-DIA-80-2003 of the Dipartimento di Informatica e Automazione (Università di Roma Tre). ARACNE Editrice, Italy

[261] Sánchez C, Sankaranarayanan S, Sipma H, Zhang T, Dill DL, Manna Z (2003) Event correlation: language and semantics. In: International conference on embedded software (EMSOFT), vol 2855. Lecture notes in computer science. Springer, pp 323–339

[262] Sankaranarayanan S, Sipma H, Manna Z (2004) Constructing invariants for hybrid systems. In: International conference on hybrid systems, vol 2993. Lecture notes in computer science. Springer, pp 539–554

[263] Sankaranarayanan S, Sipma H, Manna Z (2004) Non-linear loop invariant generation using Gröbner bases. In: International conference on principles of programming languages (POPL). ACM, pp 318–329

[264] Zhang T, Sipma HB, Manna Z (2004) Decision procedures for recursive data structures with integer constraints. In: International joint conference on automated reasoning (IJCAR), vol 3097. Lecture notes in computer science. Springer, pp 152–167

[265] Sankaranarayanan S, Sipma HB, Manna Z (2004) Constraint-based linear-relations analysis. In: Static analysis symposium (SAS), vol 3148. Lecture notes in computer science. Springer, pp 53–68

[266] Zhang T, Sipma HB, Manna Z (2004) Term algebras with length function and bounded quantifier alternation. In: International conference on theorem proving in higher order logics (TPHOLs), vol 3223. Lecture notes in computer science. Springer, pp 321–336

[267] Bradley AR, Manna Z, Sipma HB (2005) Termination of polynomial programs. In: International conference on verification, model checking, and abstract interpretation (VMCAI), vol 3385. Lecture notes in computer science. Springer, pp 113–129

[268] Sankaranarayanan S, Sipma HB, Manna Z (2005) Scalable analysis of linear systems using mathematical programming. In: International conference on verification, model checking, and abstract interpretation (VMCAI), vol 3385. Lecture notes in computer science. Springer, pp 25–41

[269] Manna Z (2005) Next generation software development. Final report DAAD 19-01-1-0723. Army Research Office (ARO)

[270] D'Angelo B, Sankaranarayanan S, Sánchez C, Robinson W, Finkbeiner B, Sipma HB, Mehrotra S, Manna Z (2005) LOLA: runtime monitoring of synchronous systems. In: International symposium on temporal representation and reasoning (TIME). IEEE, pp 166–174

[271] Bradley AR, Manna Z, Sipma HB (2005) Linear ranking with reachability. In: International conference on computer-aided verification (CAV), vol 3576. Lecture notes in computer science. Springer, pp 491–504

[272] Bradley AR, Manna Z, Sipma HB (2005) The polyranking principle. In: International colloquium on automata, languages and programming (ICALP), vol 3580. Lecture notes in computer science. Springer, pp 1349–1361

[273] Zhang T, Sipma HB, Manna Z (2005) The decidability of the first-order theory of Knuth–Bendix order. In: International conference on automated deduction (CADE), vol 3632. Lecture notes in computer science. Springer, pp 131–148

[274] Bradley AR, Manna Z, Sipma HB (2005) Termination analysis of integer linear loops. In: International conference on concurrency theory (CONCUR), vol 3653. Lecture notes in computer science. Springer, pp 488–502

[275] Sánchez C, Sipma HB, Slanina M, Manna Z (2005) Final semantics for event-pattern reactive programs. In: Conference on algebra and coalgebra in computer science (CALCO), vol 3629. Lecture notes in computer science. Springer, pp 364–378

[276] Sánchez C, Sipma HB, Subramonian V, Gill CD, Manna Z (2005) Thread allocation protocols for distributed real-time and embedded systems. In: International conference on formal techniques for distributed systems (FORTE), vol 3731. Lecture notes in computer science. Springer, pp 159–173

[277] Sánchez C, Slanina M, Sipma HB, Manna Z (2005) Expressive completeness of an event-pattern reactive programming language. In: International conference on formal techniques for distributed systems (FORTE), vol 3731. Lecture notes in computer science. Springer, pp 529–532

[278] Bradley AR, Manna Z (2005) Termination and invariance analysis of loops. In: International symposium on automated technology for verification and analysis (ATVA), vol 3707. Lecture notes in computer science. Springer, p 2

[279] Zhang T, Sipma HB, Manna Z (2005) Decision procedures for queues with integer constraints. In: Conference on foundations of software technology and theoretical computer science (FSTTCS), vol 3821. Lecture notes in computer science. Springer, pp 225–237

[280] Bradley AR, Manna Z, Sipma HB (2006) What's decidable about arrays? In: International conference on verification, model checking, and abstract interpretation (VMCAI), vol 3855. Lecture notes in computer science. Springer, pp 427–442

[281] Sankaranarayanan S, Colón M, Sipma HB, Manna Z (2006) Efficient strongly relational polyhedral analysis. In: International conference on verification, model checking, and abstract interpretation (VMCAI), vol 3855. Lecture notes in computer science. Springer, pp 111–125

[282] Sankaranarayanan S, Sipma HB, Manna Z (2006) Fixed point iteration for computing the time elapse operator. In: International conference on hybrid systems, vol 3927. Lecture notes in computer science. Springer, pp 537–551

[283] Sánchez C, Sipma HB, Manna Z, Gill CD (2006) Efficient distributed deadlock avoidance with liveness guarantees. In: International conference on embedded software (EMSOFT). ACM, pp 12–20

[284] Sánchez C, Sipma HB, Manna Z, Subramonian V, Gill CD (2006) On efficient distributed deadlock avoidance for real-time and embedded systems. In: International parallel and distributed processing symposium (IPDPS). IEEE, p 133

[285] Zhang T, Sipma HB, Manna Z (2006) Decision procedures for term algebras with integer constraints. Inf Comput 204(10):1526–1574

[286] Bradley AR, Manna Z (2006) Verification constraint problems with strengthening. In: International colloquium on theoretical aspects of computing (ICTAC), vol 4281. Lecture notes in computer science. Springer, pp 35–49

[287] Slanina M, Sipma HB, Manna Z (2006) Proving ATL* properties of infinite-state systems. In: International colloquium on theoretical aspects of computing (ICTAC), vol 4281. Lecture notes in computer science. Springer, pp 242–256

[288] Sánchez C, Sipma HB, Gill CD, Manna Z (2006) Distributed priority inheritance for real-time and embedded systems. In: International conference on principles of distributed systems (OPODIS), vol 4305. Lecture notes in computer science. Springer, pp 110–125

[289] Sánchez C, Sipma HB, Manna Z (2007) Generating efficient distributed deadlock avoidance controllers. In: International parallel and distributed processing symposium (IPDPS). IEEE, pp 1–8

[290] Slanina M, Sankaranarayanan S, Sipma H, Manna Z (2007) Controller synthesis of discrete linear plants using polyhedra. Technical report TR-2007-01. REACT, Stanford University

[291] Sánchez C, Sipma HB, Manna Z (2007) A family of distributed deadlock avoidance protocols and their reachable state spaces. In: Conference on fundamental approaches to software engineering (FASE), vol 4422. Lecture notes in computer science. Springer, pp 155–169

[292] Bradley AR, Manna Z (2007) The calculus of computation: decision procedures with applications to verification. Springer, Berlin

[293] Manna Z, Sipma HB, Zhang T (2007) Verifying balanced trees. In: Symposium on logical fundations of computer science (LFCS), vol 4514. Lecture notes in computer science. Springer, pp 363–378

[294] Bradley AR, Manna Z (2007) Checking safety by inductive generalization of counterexamples to induction. In: Conference on formal methods in computer-aided design (FMCAD). IEEE, pp 173–180

[295] Bradley AR, Manna Z (2008) Property-directed incremental invariant generation. Form Asp Comput 20(4–5):379–405

[296] Sánchez C, Slanina M, Sipma HB, Manna Z (2008) The reaction algebra: a formal language for event correlation. In: Pillars of computer science: essays dedicated to Boris (Boaz) Trakhtenbrot on the occasion of his 85th birthday, vol 4800. Lecture notes in computer science. Springer, pp 596–619

[297] Sankaranarayanan S, Sipma HB, Manna Z (2008) Constructing invariants for hybrid systems. Form Methods Syst Des 32(1):25–55

[298] Slanina M, Sipma HB, Manna Z (2008) Deductive verification of alternating systems. Form Asp Comput 20(4–5):507–560

[299] Manna Z, Peled DA (eds) (2010) Time for verification: essays in memory of Amir Pnueli, vol 6200. Lecture notes in computer science. Springer, Berlin

[300] Manna Z, Pnueli A (2010) Temporal verification of reactive systems: response. In: Time for verification: essays in memory of Amir Pnueli. Lecture notes in computer science, vol 6200. Springer, pp 279–361