Zohar Manna (1939–2018)

Nachum Dershowitz and Richard Waldinger

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.

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

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 1978, he was recruited back to Stanford as a full professor of computer science (taking Nachum along with him), 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 program verification and program 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 Section 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. His work inspired several generations of researchers. He pioneered program verification and program 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 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.”

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

- the ACM Programming Systems and Languages Award (1974, for [38]);

---

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

• 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 organization of the International Conference on Automated Deduction series) “for his pioneering research and pedagogical contributions to automated reasoning, program 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 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:


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].

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, 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).

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 programs; automated deduction; concurrent and reactive systems; real-time and hybrid systems; verification of hardware and software; and synthesis of programs. His students and colleagues dedicated their research careers to the hardest problems in automated reasoning, including program semantics, partial correctness, termination, invariant generation, program synthesis, program transformation, planning, proof methodology, temporal reasoning, natural language understanding, non-clausal 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 second-order logic to formalize the problem of termination for abstract programs and demonstrate decidable subclasses of the termination program. 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. Instead, it made stronger demands on the theorem prover, which had to be second order. Richard Lipton calls it “one of the coolest PhD theses ever.”

Zohar’s magnificent textbook, *Mathematical Theory of Computation* [39], was extraordinarily influential. It pioneered the logical analysis of programs for correctness vis-à-vis 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):

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 subprograms 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 rôle 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 programs [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 programs, 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 program 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 program transformations to deal with imperative as well as functional programs within a system dubbed DEDALUS. Eventually they moved to a purely “deductive” approach in which, to construct a program 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 program 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 program synthesis came from its use by NASA in the Amphion project, which synthesized programs 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 program that met the specification, composed of routines from the library, was extracted from the proof. That program was translated into Fortran and presented to the user. Programs synthesized by Amphion were used to analyze photographs from the Cassini mission to Saturn.

Zohar and Richard also developed a non-clausal 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 program 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.

In response, Edsger Dijkstra wrote this to Zohar:

> 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.” 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).

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)

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


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


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
Among many other contributions, Zohar and his students developed powerful methods for theory of computation (Cadiou, Chandra, Vuillemin, Shamir), program verification (Katz, Chang, Kapur, de Alfaro, Bjørner, Uribe, Sipma, Finkbeiner, Colon, Sankaranarayanan, Slanina, Bradley), program 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 (Stanford University) 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 traveling 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 center 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.

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 \textsc{TeX} by Richard’s former housemate, Evelyn Eldridge-Diaz.

When Nachum took Zohar’s course at Weizmann in 1974, \textit{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:

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.

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, \textit{joie de vivre}, and the vibrant example of both hard work and great play set by Zohar. (See Fig. \ref{fig:zohar_students}.) And Zohar’s devotion to his students is legendary.

\footnote{https://rjlipton.wordpress.com/2014/11/12/three-in-the-room}
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 accomplishments, and was devoted to all who came into his orbit.

He will be missed sorely.

Publication List

Zohar Manna (1939–2018) 11


16 Nachum Dershowitz and Richard Waldinger


\[\text{Zohar Manna (1939–2018)}\]


