

Mordechai (Moti) Ben-Ari

Education

- Ph.D. Tel Aviv University, Mathematics (Computer Science), 1981.
Thesis: *Complexity of Proofs and Models in Programming Logics*.
Advisors: Amir Pnueli and Zvi Galil.
- M.Sc. Tel Aviv University, Mathematics (Computer Science), 1977.
Thesis: *New Translations of Program Schemes*.
Advisors: Amir Pnueli and Zvi Galil.
- B.Sc., Mathematics, Massachusetts Institute of Technology, 1970.

Employment

- 2019–: Department of Science Teaching, Weizmann Institute of Science. Professor (Emeritus).
- 1995–2018: Department of Science Teaching, Weizmann Institute of Science. Professor.
- 1993–95: Mavo Software Ltd. Founder and general manager.
- 1981–92: Rafael. Research scientist and software section head.
- 1972–81: Tel Aviv University. Systems programmer, teaching assistant, instructor.

Visiting academic positions

- 2005: Helsinki University of Technology.
- 2002: University of Joensuu.
- 1988: Brandeis University.

Awards

- ACM SIGAda Outstanding Ada Community Contribution Award, 2013.
- ACM Distinguished Educator, 2009.
- ACM SIGCSE Award for Outstanding Contributions to Computer Science Education, 2004.

Editorial board or associate editor (retired in 2015):

- ACM Transactions on Computing Education.
- ACM Inroads.
- Computer Science Education.
- IEEE Transactions on Learning Technologies.

Referee and/or program committee member

- Special Interest Group in Computer Science Education (SIGCSE) Symposium.
- Conference on Innovation and Technology in Computer Science Education (ITiCSE).
- Koli Calling: Finnish/Baltic Sea Conference on Computer Science Education.
- International Computing Education Research (ICER) Conference.

Keynote speeches at conferences

- *In Defense of Programming*. 20th Annual Conference on Innovation and Technology in Computer Science Education, Vilnius, Lithuania, 8 July 2015.
- *Non-myths About Programming*. 6th International Computing Education Research Conference, Århus, Denmark, 9 August 2010.
- *Objects Never: Well, Hardly Ever*. XIV Jornadas de Enseñanza Universitaria de Informática, Granada, Spain, 9 July, 2008.
- *The Effect of the Jeliot Animation System on Learning Elementary Programming*. 4th Greek Conference on the Didactics of Informatics. Patras, Greece, 29 March, 2008.
- *The Invisible Programmers*. Conference on Methods, Materials and Tools for Programming Education, Tampere, Finland, 4 May, 2006.
- *The Concorde Doesn't Fly Anymore*. Thirty-Sixth SIGCSE Technical Symposium on Computer Science Education, St. Louis, MO, 25 March 2005.
- *From Theory to Experiment to Practice in CS Education*. 2nd Annual Finnish/Baltic Sea Conference on Computer Science Education, Koli, Finland, 19 October, 2002.

Research Grants

- The Influence of Robotics Activities on Students' Attitudes and Intentions Towards Science and Technology. Israel Science Foundation (ISF), \$136,908, 2013–2016.
- Computer Science Education in Middle Schools: Effectiveness and Preparedness. Israel Science Foundation (ISF), \$88,378, 2009–2012.

Development Grants from the Israeli Ministry of Education

- National Teachers Center in Computer Science, 2000–2005 (jointly with the Technion).
- Visualization tools for teaching models of computation, 2006.
- Visualization tools for teaching data structures, 2004.
- Concurrent and distributed computation, 1997–2002, 2003.
- Object-oriented programming and Java in introductory computer science, 2000.
- Foundations of computer science 1 & 2, 1995–2000.

Selected courses taught

- Mathematics of Programming, Rothschild-Weizmann Program for Excellence in Science Teaching, Weizmann Institute of Science, 2011, 2013, 2015, 2017.
- Philosophy of Education, Weizmann Institute of Science, 2009, 2011.
- Thesis Seminar, Weizmann Institute of Science, 2006, 2008, 2010, 2012.
- Philosophy and History of Science in Science Teaching, Weizmann Institute of Science, 2002, 2004, 2006, 2015.
- Seminar in the Theory of Computer Science Education, Helsinki University of Technology, 2005.

Graduate Students Supervised¹

Postdoctoral Fellows

1. Orni Meerbaum-Salant, 2010-12.
Currently: Project director, Davidson Institute of Science Education.
2. Michal Armoni, 2007–2009.
Currently: Senior scientist now under review for tenure in the department.

Ph.D. Students

1. Fatima Kaloti-Hallak. *The Effect of Robotics Activities on Students' Learning and Attitudes*, 2015. Joint supervisor: Michal Armoni.
Currently: Assistant professor, Bir Zeit University, Palestinian Authority.
2. Rivka Taub. *The Effect of Computer Science on Physics Learning in a Computational Environment*, 2014. Joint supervisor: Michal Armoni.
Currently: Director of the education department, Hevel Yavne regional council.
3. Jan Lönnberg. *Understanding and Debugging Concurrent Programs through Visualisation*, 2012. Department of Computer Science and Engineering, Aalto University, Finland. Joint supervisor: Lauri Malmi.
Currently: Senior software engineer. RedLynx.
4. Ronit Ben-Bassat Levy. *Teaching Computer Science with Animation: Attitudes and Ways of Experiencing*, 2009.
Currently: Senior teacher of mathematics and computer science, and part-time researcher in the department.

¹At the Department of Science Teaching of the Weizmann Institute of Science unless otherwise noted.

5. Niko Myller. *Collaborative Software Visualization for Learning: Theory and Applications*. Department of Computer Science, University of Joensuu, Finland, 2009. Joint supervisor: Erkki Sutinen.
Currently: Chief technical officer, SN4 International Oy.
6. Noa Ragonis. *Teaching Object-Oriented Programming to Novices*, 2004.
Currently: Head of the instructional development center and senior lecturer, Beit Berl College; adjunct senior lecturer, Department of Education in Technology and Science, Technion.
7. Cecile Yehezkel. *A Visualization Environment for Computer Architecture*, 2004. Joint supervisor: Tommy Dreyfus.
Currently: Head of instructional laboratories, School of Engineering, Bar-Ilan University.
8. Yifat Ben-David Kolikant. *Understanding Concurrency: The Process and the Product*, 2003.
Currently: Senior lecturer (tenured), School of Education, Hebrew University of Jerusalem.

M.Sc. Students

1. Mor Friebronn Yesharim. *What do elementary-school students learn in computer science classes with robotics?*, 2018.
2. Mikko Vinni. *Use of Cognitive Dimensions in Software Development: The Case Study of a Content Editor for a Mobile Guide in a Museum*, School of Computing, University of Eastern Finland, 2012. Joint supervisor: Teemu Laine.
3. Fatima Kaloti-Hallak. *Learning Programming Concepts Using Scratch at the Middle-School Level*, 2010. Joint supervisor: Michal Armoni.
4. Rivka Taub. *CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS*, 2010. Joint supervisor: Michal Armoni.
5. Trishank Karthik Kuppusamy. *The Pomegranate Promela Compiler*. Department of Computer Science, New York University, 2009. Joint supervisor: Edmond Schonberg.
6. Gil Ebel. *The Effect of Program Visualization on the Attention-Directing Characteristics of the Learner*, 2006.
7. Shmuel Schwartz. *Using State Diagrams for Understanding Correctness in Concurrent Programming*, 2006.
8. Andrés Moreno García. *The Design and Implementation of Intermediate Codes for Software Visualization*. Department of Computer Science. University of Joensuu, Finland, 2004. Joint supervisor: Erkki Sutinen.

9. Yevgeniya Kulikova. *Roles of Variables in Teaching Functional Programming*. Department of Computer Science, University of Joensuu, Finland, 2004. Joint supervisor: Jorma Sajaniemi.
10. Maxim Mozgovoy. *Concurrent Program Verifier: A Tool for Teaching Concurrent Programming*. Department of Computer Science, University of Joensuu, Finland, 2004. Joint supervisor: Erkki Sutinen.
11. Niko Myller. *The Fundamental Design Issues of Jeliot 3*. Department of Computer Science, University of Joensuu, Finland, 2004. Joint supervisor: Erkki Sutinen.
12. Tzipora Yeshno. *Teaching an Explicit Conceptual Model As a Means to Improve the Work with Computer Applications*, 2003.
13. Ronit Ben-Bassat Levy. *The Use of Animation as an Educational Tool*, 2002.
14. Yekaterina Sedletzky. *Formal Verification of Distributed Algorithms*. Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 2000. Joint supervisor: Amir Pnueli.
15. Yakov Persky. *Re-engineering a Concurrency Simulator*. Department of Computer Science, Tel-Aviv University, 1999.
16. Yoav Tsruya. *A Distributed Programming Environment in Ada95/Java*. Department of Computer Science, Tel-Aviv University, 1998.
17. Noa Ragonis. *Introduction to Expert Systems: Development and Evaluation of a Computer Science Curriculum*, 1997. Joint supervisors: Ehud Shapiro, Zehava Scherz.
18. Ophira Statman. *A Prolog Proof Checker for Temporal Logic*. Department of Computer Science, Technion, 1986.

List of Publications

Textbooks

1. M. Ben-Ari. *Principles of Concurrent Programming*. Prentice-Hall International, 1982. (Translated into French, German, Polish, Japanese.)
2. M. Ben-Ari. *Principles of Concurrent and Distributed Programming*. Prentice-Hall International, 1990. (Translated into Italian.)
3. M. Ben-Ari. *Mathematical Logic for Computer Science*. Prentice-Hall International, 1992. (Translated into Italian.)
4. M. Ben-Ari. *Understanding Programming Languages*. John Wiley, 1995. (Translated into Russian.)
5. M. Ben-Ari. *Ada for Software Engineers*. John Wiley & Sons, 1998.
6. M. Ben-Ari, O. Lichtenstein, H. Mahlav, N. Reich. *Foundations of Computer Science 2*. Weizmann Institute of Science, 1996–1999 (in Hebrew; translated into Arabic).
7. M. Ben-Ari. *Mathematical Logic for Computer Science (Second Revised Edition)*. Springer, 2001. (Translated into Polish.)
8. Y. Ben-David Kolikant, M. Ben-Ari, H. Mahlav, A. Tikvati. *Concurrent and Distributed Computation*. Weizmann Institute of Science, 1998–2002 (in Hebrew).
9. M. Ben-Ari. *Just a Theory: Exploring the Nature of Science*. Prometheus, 2005.
10. M. Ben-Ari. *Principles of Concurrent and Distributed Programming (Second Edition)*. Addison-Wesley, 2006. (Translated into Polish.)
11. M. Ben-Ari. *Principles of the Spin Model Checker*, Springer, 2008. (Translated into Japanese.)
12. M. Ben-Ari. *Ada for Software Engineers (Second Edition with Ada 2005)*. Springer, 2009.
13. M. Armoni, M. Ben-Ari. *Computer Science Concepts in Scratch*, Weizmann Institute of Science, 2010 (Hebrew and English).
14. M. Ben-Ari. *Mathematical Logic for Computer Science (Third Edition)*. Springer, 2012.
15. M. Ben-Ari, F. Mondada. *Elements of Robotics*, Springer, 2017.
Download at <http://www.springer.com/gp/book/9783319625331/>.

Refereed Journals

16. M. Ben-Ari. Ianov pushdown schemes are contained in boolean recursive schemes. *Acta Informatica* 10(1977), 117–125.
17. M. Ben-Ari. On transposing large $2^n \times 2^n$ matrices. *IEEE Transactions on Computers* C28 (1979), 72–75.
18. M. Ben-Ari. Why you should not time-share. *Software—Practice and Experience* 9(1979), 339–340.
19. M. Ben-Ari. A simplified proof that regular resolution is exponential. *Information Processing Letters* 10(1980), 96–98.
20. M. Ben-Ari. Comments on “Tautology testing with a generalized matrix reduction method”. *Theoretical Computer Science* 11(1980), 341.
21. M. Ben-Ari. Cheap concurrent programming. *Software—Practice and Experience* 11(1981), 1261–1264.
22. M. Ben-Ari, J.Y. Halpern, A. Pnueli. Deterministic propositional dynamic logic: Finite models, complexity, and completeness. *Journal of Computer and System Sciences* 25(2), 1982, 402–417.
23. M. Ben-Ari, Z. Manna, A. Pnueli. The temporal logic of branching time. *Acta Informatica* 20, 1983, 207–226.
24. M. Ben-Ari. Algorithms for on-the-fly garbage collection. *ACM Transactions on Programming Languages and Systems* 6(3), 1984, 333–344.
25. M. Ben-Ari. How to solve the Santa Claus problem. *Concurrency: Practice and Experience* 10(6), 1998, 485–496.
26. M. Ben-Ari. Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching* 20(1), 2001, 45–73.
27. M. Ben-Ari. Interactive execution of distributed algorithms. *ACM Journal on Educational Resources in Computing*, 1(2), 2001.
28. M. Ben-Ari. Theory-guided technology in computer science. *Science & Education* 10(5), 2001, 477–484.
29. R. Ben-Bassat Levy, M. Ben-Ari, P.A. Uronen. The Jeliot 2000 program animation system. *Computers & Education* 40(1), 2003, 1–15.
30. M. Ben-Ari. The NOMA of Yishayahu Leibowitz. *Science & Education* 12(7), 2003, 719–723.

31. M. Ben-Ari. On random numbers and design. *Science & Education* 13(3), 2004, 235–241.
32. M. Ben-Ari. Situated learning in computer science education. *Computer Science Education* 14(2), 2004, 85–100.
33. M. Ben-Ari. Situated learning in this high-technology world. *Science & Education* 14(3-5), 2005, 367–376.
34. N. Ragonis, M. Ben-Ari. A long-term investigation of the comprehension of OOP concepts by novices. *Computer Science Education* 15(3), 2005, 203–221.
35. M. Ben-Ari, T. Yeshno. Conceptual models of software artifacts. *Interacting with Computers* 18 (6), 2006, 1336–1350.
36. J. Sajaniemi, M. Ben-Ari, P. Byckling, P. Gerdt, Y. Kulikova. Roles of variables in three programming paradigms. *Computer Science Education* 16(4), 2006, 261–279.
37. C. Yehezkel, M. Ben-Ari, T. Dreyfus. The contribution of visualization to learning computer architecture. *Computer Science Education* 17(2), 2007, 117–127.
38. Y. Ben-David Kolikant, M. Ben-Ari. Fertile zones of cultural encounter. *Journal of the Learning Sciences* 17(1), 2008, 1–32.
39. R. Ben-Bassat Levy, M. Ben-Ari. Adapting and merging methodologies in doctoral research. *Computer Science Education* 19(2), 2009, 51–67.
40. M. Armoni, M. Ben-Ari. The concept of nondeterminism: Its development and implications for education. *Science & Education* 18(8), 2009, 1005–1030. Reprinted in: *inroads: SIGCSE Bulletin* 41(2), 2009, 141–160.
41. N. Myller, R. Bednarik, M. Ben-Ari, E. Sutinen. Extending the engagement taxonomy: Software visualization and collaborative learning. *ACM Transactions on Computing Education*, 2009, 7:1–7:27.
42. M. Ben-Ari, R. Bednarik, R. Ben-Bassat Levy, G. Ebel, A. Moreno, N. Myller, E. Sutinen. A decade of research and development on program animation: The Jeliot experience. *Journal of Visual Languages and Computing*, 22(5), 375–384, 2011.
43. R. Taub, M. Ben-Ari, M. Armoni. CS Unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education*, 12(2), 8:1–8:29, 2012.
44. O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 2013, 239–264.
45. M. Armoni, O. Meerbaum-Salant, M. Ben-Ari. From Scratch to “Real” Programming. *ACM Transactions on Computing Education*, 14(4), article 25, 2015.

46. R. Taub, M. Ben-Ari, M. Armoni. The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87, 10–23, 2015.
47. F. Kaloti-Hallak, M. Armoni, M. Ben-Ari. The effectiveness of robotics competitions on students' learning of computer science, *Olympiads in Informatics* 9, 89–112, 2015.
48. M. Ben-Ari. LearnSAT: A SAT solver for education. *Journal of Open Source Software*, 3(24), 639, 2018. <https://doi.org/10.21105/joss.00639>.
49. M. Friebronn-Yesharim, M. Ben-Ari. Teaching computer science concepts through robotics to elementary school children. *International Journal of Computer Science Education in Schools* 2(3), 2018.
50. R. Taub, M. Ben-Ari, M. Armoni. Physics conceptual understanding in a computational science course. *Journal of Computational Science Education* 9(2), 2018.
51. F. Kaloti-Hallak, M. Armoni, M. Ben-Ari. The effect of robotics activities on learning the engineering design process. *Informatics in Education* 18(1), 105–129, 2019.

Conference Proceedings

52. M. Ben-Ari, Z. Manna, A. Pnueli. The temporal logic of branching time. *Eighth ACM Symposium on Principles of Programming Languages*. Williamsburg, VA, 1981, 164–176.
53. M. Ben-Ari, J.Y. Halpern, A. Pnueli. Finite models for deterministic propositional dynamic logic. *Eighth International Colloquium on Automata, Languages and Programming*. Haifa, Israel, 1981, 249–263.
54. M. Ben-Ari. On-the-fly garbage collection: New algorithms inspired by program proofs. *Ninth International Colloquium on Automata, Languages, and Programming*. Århus, Denmark, 1982, 14–22.
55. M. Ben-Ari. Reverse engineering into Ada. *Workshop on Software Methodologies in Ada*, Tel-Aviv, Israel, 1983.
56. H. Yashinsky, M. Ben-Ari. Ada for military microcomputers. *Second Israel Conference on Computer Systems Engineering and Software Engineering*. Tel-Aviv, Israel, 1987 (in Hebrew).
57. M. Ben-Ari. Ada requirements for small real-time systems. *Third International Workshop on Real-Time Ada Issues*. Nemaquin Woodlands, PA, 1989, 159–165.
58. M. Ben-Ari. Experience teaching object-oriented programming in Ada. *Symposium on Teaching Object Technology*. Santa Barbara, CA, 1996.
59. M. Ben-Ari. Using inheritance to implement concurrency. *Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education*. Philadelphia, PA, 1996, 180–184.

60. M. Ben-Ari. Distributed algorithms in Java. *Second SIGCSE Conference on Integrating Technology into Computer Science Education*. Uppsala, Sweden, 1997, 62-64.
61. M. Ben-Ari. The software factory. *Psychology of Programming Interest Group Tenth Annual Workshop*. Milton Keynes, UK, 1998, 89-91.
62. M. Ben-Ari. Constructivism in computer science education. *Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education*. Atlanta, GA, 1998, 257-261.
63. M. Ben-Ari. Synchronizing multiple clients and servers. *Ada-Europe International Conference on Reliable Software Technologies*. Uppsala, Sweden, Lecture Notes in Computer Science 1411, 1998, 40-51.
64. Y. Persky, M. Ben-Ari. Re-engineering a concurrency simulator. *Third SIGCSE Conference on Integrating Technology into Computer Science Education*. Dublin, Ireland, 1998, 185-188.
65. Y. Tzruya, M. Ben-Ari. A portable implementation of the distributed systems annex in Java. *SIGADA '98: Ada in Context*. Washington, DC, 1998, 204-211.
66. M. Ben-Ari. Bricolage forever! *Eleventh Annual Workshop of the Psychology of Programming Interest Group*. Leeds, UK, 1999, 53-57.
67. M. Ben-Ari, S. Silverman. DPLab: An environment for distributed programming. *Fourth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Cracow, Poland, 1999, 91-94.
68. M. Ben-Ari, Y. Ben-David Kolikant. Thinking parallel: The process of learning concurrency. *Fourth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Cracow, Poland, 1999, 13-16.
69. M. Ben-Ari. Theory-guided technology in computer science. *Fifth International History, Philosophy and Science Teaching Conference*. Como, Italy, 1999.
70. R. Ben-Bassat Levy, M. Ben-Ari, P.A. Uronen. An extended experiment with Jeliot 2000. *First Program Visualization Workshop*. Porvoo, Finland, 2000, 131-140.
71. Y. Ben-David Kolikant, M. Ben-Ari, S. Pollack. The anthropology of semaphores. *Fifth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Helsinki, Finland, 2000, 21-24.
72. E. Sedletsky, A. Pnueli, M. Ben-Ari. Formal verification of the Ricart-Agrawala algorithm. *Foundations of Software Technology and Theoretical Computer Science New Delhi, India, Lecture Notes in Computer Science 1974*, 2000, 325-335.
73. T. Yeshno, M. Ben-Ari. Salvation for bricoluers. *Thirteenth Annual Workshop of the Psychology of Programming Interest Group*. Bournemouth, UK, 2001, 225-235.

74. M. Ben-Ari. The argument for design. *Sixth International History, Philosophy and Science Teaching Conference*. Denver, CO, 2001.
75. M. Ben-Ari, N. Myller, E. Sutinen, J. Tarhio. Perspectives on program animation with Jeliot. *Software Visualization: International Seminar*. Dagstuhl Castle, Germany, Lecture Notes in Computer Science 2269, 2002, 31–45.
76. N. Ragonis, M. Ben-Ari. Teaching constructors: A difficult multiple choice. *Sixth Workshop on Pedagogies and Tools for Learning Object-Oriented Concepts, Sixteenth European Conference on Object-Oriented Programming*. Malaga, Spain, 2002.
77. M. Ben-Ari, N. Ragonis, R. Ben-Bassat Levy. A vision of visualization in teaching object-oriented programming. *Second Program Visualization Workshop*. Hornstrup-Centret, Denmark, 2002, 83–89.
78. M. Ben-Ari. From theory to experiment to practice in CS education. *Kolin Kolistelut - Koli Calling: Second Annual Finnish/Baltic Sea Conference on Computer Science Education*. Koli, Finland, October, 2002.
79. M. Ben-Ari. Situated learning in this high-technology world. *Seventh International History, Philosophy and Science Teaching Conference*. Winnipeg, Canada, 2003.
80. A. Tikvati, M. Ben-Ari, Y. Ben-David Kolikant. Virtual trees for the Byzantine Generals algorithm. *Thirty-Fifth SIGCSE Technical Symposium on Computer Science Education*, Norfolk, VA, 2004, 392–396.
81. A. Moreno, N. Myller, E. Sutinen, M. Ben-Ari. Visualizing programs with Jeliot 3. *Conference on Advanced Visual Interfaces*, Gallipoli, Italy, 2004, 373–376.
82. M. Ben-Ari, J. Sajaniemi. Roles of variables from the perspective of computer science educators. *Ninth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Leeds, UK, 2004, 52–56.
83. S. Pollack, M. Ben-Ari. Selecting a visualization system. *Third Program Visualization Workshop*. Warwick, UK, 2004, 121–126.
84. C. Yehezkel, M. Ben-Ari, T. Dreyfus. Inside the computer: Visualization and mental models. *Third Program Visualization Workshop*. Warwick, UK, 2004, 77–80.
85. C. Yehezkel, M. Ben-Ari, T. Dreyfus. Computer architecture and mental models. *Thirty-Sixth SIGCSE Technical Symposium on Computer Science Education*, St. Louis, MO, 2005, 101–105.
86. N. Ragonis, M. Ben-Ari. On understanding the statics and dynamics of object-oriented programs. *Thirty-Sixth SIGCSE Technical Symposium on Computer Science Education*, St. Louis, MO, 2005, 226–230.
87. M. Ben-Ari. Whose final hour? The rise of naive egocentric catastrophism. *Eighth International History, Philosophy and Science Teaching Conference*. Leeds, UK, 2005.

88. S. Schwarz, M. Ben-Ari. Why don't they do what we want them to do? *Eighteenth Psychology of Programming Group Workshop*. Brighton, UK, 2006, 266–274.
89. G. Ebel, M. Ben-Ari. Affective effects of program visualization. *Second International Computing Education Research Conference*. Canterbury, UK, 2006, 1–5.
90. R. Ben-Bassat Levy, M. Ben-Ari. We work so hard and they don't use it: Acceptance of software tools by teachers. *Twelfth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Dundee, UK, 2007, 246–250.
91. M. Armoni, N. Lewenstein, M. Ben-Ari. Teaching students to think nondeterministically. *Thirty-Ninth SIGCSE Technical Symposium on Computer Science Education*, Portland, OR, 2008, 4–8.
92. M. Ben-Ari. The effect of the Jeliot animation system on learning elementary programming. *Fourth Greek Conference on the Didactics of Informatics*. Patras, Greece, March, 2008.
93. R. Ben-Bassat Levy, M. Ben-Ari. Perceived behavior control and its influence on the adoption of software tools. *Thirteenth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Madrid, Spain, 2008, 169–173.
94. R. Ben-Bassat Levy, M. Ben-Ari. A survey of research on the Jeliot program animation system. *Fourth Annual Chais Conference on Instructional Technologies Research: Learning in the Technological Era*, 2009.
95. R. Taub, M. Ben-Ari, M. Armoni. The effect of CS Unplugged on middle-school students' views of CS. *Fourteenth Conference on Innovation and Technology in Computer Science Education*, Paris, France, 2009, 99–103.
96. M. Ben-Ari. Teaching concurrency and model checking. *16th International SPIN Workshop on Model Checking of Software*, Grenoble, France, 2009, 6–11.
97. O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Learning computer science concepts with Scratch. *6th International Computing Education Research Conference*, Århus, Denmark, 2010, 69–76.
98. O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Habits of programming in Scratch. *Sixteenth Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, 2011, 168–172.
99. J. Lönnberg, M. Ben-Ari, L. Malmi. Java replay for dependence-based debugging. *Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging (PAD-TAD-IX)*, Toronto, ON, 2011, 15–25.
100. J. Lönnberg, M. Ben-Ari, L. Malmi. Visualising concurrent programs with dynamic dependence graphs. *6th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, Williamsburg, VA, 2011, 1–4.

101. J. Lönnberg, L. Malmi, M. Ben-Ari. Evaluating a visualisation of the execution of a concurrent program. *11th Koli Calling: International Conference on Computing Education Research*, Koli, Finland, 2011, 39–48.
102. R. Taub, M. Armoni, M. Ben-Ari. The contribution of computer science to learning computational physics. *6th International Conference on Informatics in Schools: Situation, Evolution and Perspectives*, Oldenburg, Germany, LNCS 7780, pp. 127–137, 2013.
103. M. Ben-Ari. LearnSAT: A SAT solver for education. *16th Int. Conf. on Theory and Applications of Satisfiability Testing*, Helsinki, Finland, 2013, 403–407.
104. R. Taub, M. Armoni, M. Ben-Ari. Abstraction as a bridging concept between computer science and physics. *9th Workshop in Primary and Secondary Computing Education*, Berlin, Germany, 16–19 2014.
105. S. Magnenat, J. Shin, F. Riedo, R. Siegwart, M. Ben-Ari. Teaching a core CS concept through robotics. *Nineteenth Conference on Innovation and Technology in Computer Science Education*, Uppsala, Sweden, 2014, 315–320.
106. S. Magnenat, M. Ben-Ari, S. Klinger, R. W. Sumner. Enhancing robot programming with visual feedback and augmented reality. *Twentieth Conference on Innovation and Technology in Computer Science Education*, Vilnius, Lithuania, 2015, 153–158.
107. R. Ben-Bassat Levy and M. Ben-Ari. Robotics activities—Is the investment worthwhile? *Eight International Conference on Informatics in Schools*, Ljubljana, Slovenia, 2015, 22–31.
108. F. Kaloti-Hallak, M. Armoni, M. Ben-Ari. Students' Attitudes and Motivation During Robotics Activities. *Tenth Workshop in Primary and Secondary Computing Education*, London, 2015.
109. M. Friebroon-Yesharim, M. Ben-Ari. Teaching Robotics Concepts to Elementary School Children. *8th International Conference on Robotics in Education*, Sofia, Bulgaria, 2017, 26–28.
110. R. Ben-Bassat Levy, M. Ben-Ari. The Evaluation of Robotics Activities for Facilitating STEM Learning, *8th International Conference on Robotics in Education*, Sofia, Bulgaria, 2017, 132–137.

Conference Posters

111. N. Ragonis, Z. Scherz, M. Ben-Ari. Development, implementation and evaluation of a course in expert systems for high-school students. *Third SIGCSE Conference on Integrating Technology into Computer Science Education*. Dublin, Ireland, 1998.

112. M. Ben-Ari, J. Sajaniemi. *Roles of Variables From the Perspective of Computer Science Educators*. Technical Report A-2003-6, University of Joensuu, 2003.
113. A. Moreno, N. Myller, M. Ben-Ari, E. Sutinen. Program animation in Jeliot 3. *Ninth SIGCSE Conference on Integrating Technology into Computer Science Education*. Leeds, UK, 2004.
114. M. Ben-Ari. A suite of tools for teaching concurrency. *Ninth SIGCSE Conference on Integrating Technology into Computer Science Education*. Leeds, UK, 2004.
115. M. Ben-Ari. ConcurrencySuite: Teaching concurrency and nondeterminism with Spin. *Jornadas de Enseñanza Universitaria de Informática*. Granada, Spain, July, 2008.
116. A.L.A. Moth, J. Villadsen, M. Ben-Ari. SyntaxTrain: Relieving the Pain of Syntax. *Sixteenth SIGCSE Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, 2011.
117. R. Taub, M. Ben-Ari, M. Michal Armoni. The effect of computer science on the learning of computational physics. *Nineteenth SIGCSE Conference on Innovation and Technology in Computer Science Education*, Uppsala, Sweden, 2014,.

Reviewed Papers

118. M. Ben-Ari, A. Yehudai. A methodology for modular use of Ada. *SIGPLAN Notices* 16(12), 1981, 22–26.
119. M. Ben-Ari. The case for full Ada. *Ada Letters* II(3), 1982, 34–37.
120. M. Ben-Ari. An extremely pure logic program. *Logic Programming Letters* 3(1), 1989, 6–7.
121. M. Ben-Ari. Signaling from within interrupt handlers. *Ada Letters* X(1), 1990, 100–103.
122. M. Ben-Ari. Ada requirements for small real-time systems. *Ada Letters* X(4), 1990, 159–165.
123. M. Ben-Ari. Structure exits, not loops. *SIGCSE Bulletin*, 28(3), 1996, 51–54,59.
124. M. Ben-Ari. An improved program for teaching the Towers of Hanoi. *Aspects of Teaching Computer Science*, 7, 1996, 48–49 (in Hebrew).
125. M. Ben-Ari, N. Reich. Recursion: From drama to program. *Aspects of Teaching Computer Science*, 7, 1996, 45–47 (in Hebrew).
126. M. Ben-Ari. Parameter passing. *Aspects of Teaching Computer Science*, 6, 1996, 19-22 (in Hebrew).

127. M. Ben-Ari, K. Henney. A critique of the Advanced Placement C++ subset. *SIGCSE Bulletin* 29(2), 1997, 7–10.
128. M. Ben-Ari. Recursion: From drama to program. *Journal of Computer Science Education* 11(3), 1997, 9–12.
129. M. Ben-Ari. Dispatching on the function result. *Ada Letters* XVIII(4), 1998, 101–106.
130. M. Ben-Ari. The bug that destroyed a rocket. *Aspects of Teaching Computer Science*, 10, 1998, 20–22 (in Hebrew). *Journal of Computer Science Education* 13(2), 1999, 15–16.
131. M. Ben-Ari. Teaching object-oriented programming in Ada. *Journal of Object-Oriented Programming* 11(6), 1998, 39–45.
132. M. Ben-Ari. Parameter passing in Pascal. *Journal of Computer Science Education* 12(1-2), 1998, 27–28.
133. M. Ben-Ari, Alan Burns. Extreme interleavings. *IEEE Concurrency* 6(3), 1998, 90.
134. M. Ben-Ari. How to get a good review. *SIGCSE Bulletin* 32(2), 2000, 4–6.
135. M. Ben-Ari. Program visualization in theory and practice. *Informatik / Informatique* 2, 2001, 8–11.
136. M. Ben-Ari. Abstract data types in C. *Aspects of Teaching Computer Science*, June 2002, 49–51 (in Hebrew).
137. M. Ben-Ari. How did they get to the moon without PowerPoint? *Tietojenkäsittelytiede*, 20, 2003, 8–15 (Finnish translation; the English version is on my website).
138. Y. Ben-David Kolikant, M. Ben-Ari. Teaching computer science as fertile zones of cultural encounter. *Aspects of Teaching Computer Science*, June 2004, 21–29 (in Hebrew).
139. M. Ben-Ari. Why was there only one Japan? In Nigel Sanitt (ed.). *Motivating Science: Science Communication from a Philosophical, Educational and Cultural Perspective*, Pantaneto Press, 2005, 203–205.
140. M. Ben-Ari. Minesweeper as an NP-complete problem. *SIGCSE Bulletin* 37(4), 2005, 39–40.
141. M. Ben-Ari. Whose final hour? The problem of naive egocentric catastrophism in doomsayers and catastrophists. *Skeptic* 12(3), 2006, 40–49. Hebrew version in: *Kriat Benayim* 13, 2008, 9–15.
142. M. Ben-Ari. McKinley's Amazon. *SIGCSE Bulletin* 38(4), 2006, 75–77.
143. M. Ben-Ari. The end of science revisited: The case for incrementalism in the future of science. *Skeptic* 13(2), 2007, 20–27.

144. M. Ben-Ari. A primer on model checking. *ACM Inroads* 1(1), 2010, 40–47.
145. M. Ben-Ari. Objects never? Well, hardly ever! *Communications of the ACM* 53(9), 32–35, 2010. Hebrew version in *Aspects of Teaching Computer Science*, June, 2010, 31–37.
146. M. Ben-Ari. Loop constructs in Scratch. *ACM Inroads* 2(1), 2011, 27–28. Hebrew version in *Aspects of Teaching Computer Science*, 34, 2011, 12–13.
147. M. Ben-Ari. Non-myths about programming. *Communications of the ACM*, 54(7), 2011, 35–37. Hebrew version in *Aspects of Teaching Computer Science*, January, 2012. 7–12.
148. M. Ben-Ari, F. Kaloti-Hallak. Demonstrating random and parallel algorithm with Spin. *ACM Inroads* 3(3), 2012, 36–38.
149. M. Ben-Ari. Visualizaton of Programming. In: Kadijevich Dj.M., Angeli, C., Schulte, C. (Eds.) *Improving Computer Science Education*, Routledge, 2013.
150. M. Ben-Ari. MOOCs on Introductory Programming: A Travelogue. *ACM Inroads* 4(2), 2013, 58–61.

Learning Materials

151. (Academic supervision) R. Ben-Bassat Levy. *Program Animation in Jeliot*, 2000.
152. (Academic supervision) S. Pollack. *Algorithm Visualization in MatrixPro*, 2004.
153. M. Ben-Ari. *Minesweeper is NP Complete*, 2005.
154. (Academic supervision) S. Schwarz. *Visualization of Computational Models in JFLAP*, 2006.
155. M. Ben-Ari. *Compile and Runtime Errors in Java*, 2006.
156. M. Ben-Ari. *Learning Objects for Visualization: Programming in Java Using Jeliot*, 2007.
157. M. Ben-Ari. *First Steps in Robotics with the Thymio-II Robot*, 2013-15. (Translated into French, German and Italian.)
158. M. Ben-Ari. *Projects for the Thymio-II Robot in the Aseba Studio Environment*, 2013-15.
159. M. Ben-Ari. *Scratch Simulations of the Thymio Robot*, 2015.
160. M. Ben-Ari. *The Many Guises of Induction*, 2016–17.

Pedagogical Software Tools

161. M. Ben-Ari. *CP: A Concurrency Simulator*, 1974–80, 1996–2000.
162. (Academic supervision) Shawn Silverman. *DPLab: An environment for distributed programming*, 1998.
163. (Academic supervision with Shmuel Sagiv, Tel Aviv University) Yoav Tzruya. *A Distributed Programming Environment for Ada95/Java*, 1998.
164. (Academic supervision with Amiram Yehudai, Tel Aviv University) Yakov Persky. *SimAda Concurrency Simulator*, 1999.
165. M. Ben-Ari. *DAJ: Interactive Learning of Distributed Algorithms*, 2001, 2005.
166. (Academic supervision with Erkki Sutinen, University of Joensuu) Niko Myller, Andrés Moreno García, Ronit Ben-Bassat Levy, Roman Bednarik, Noa Ragonis. *Jeliot: A Program Animation System for Java*, 1999–2009.
167. M. Ben-Ari. *jBACI: Development Environment and Simulator for Concurrency*, 2000–04.
168. M. Ben-Ari. *jSpin: Development Environment for the Spin Model Checker*, 2004–07.
169. M. Ben-Ari. *VN: Visualizing Nondeterminism*, 2006.
170. (Academic supervision) Mikko Vinni. *iDot: Incremental Viewer for Dot Graphs*, 2006.
171. (Academic supervision) Maxim Mozgovoy. *CPV: Concurrent Program Verifier*, 2006.
172. M. Ben-Ari. *SpinSpider: Automatic Generation of State Transition Diagrams*, 2007.
173. (Academic supervision) Oshrit Feder. *Jeliot Support for Python*, 2008.
174. (Academic supervision with Edmond Schonberg, New York University) Trishank Karthik Kuppasamy. *The Pomegranate Promela Compiler*, 2009.
175. M. Ben-Ari. *The Erigone Model Checker*, 2009–11.
176. M. Ben-Ari. *EUI: Development Environment for Erigone*, 2009.
177. M. Ben-Ari. Scratch-based tools: Karel the Robot, Valentino the Robot, Computer Science Unplugged.
178. (Academic supervision with Lauri Malmi, Aalto University, Finland) Jan Lönnberg. *Atropos: Dependence-Based Replay for Debugging Concurrent Java Programs*, 2011.
179. (Academic supervision with Jørgen Villadsen, Technical University of Denmark) Andreas L. A. Moth. *SyntaxTrain: Interactive Display of Syntax Errors*, 2011.
180. (Academic supervision with Jørgen Villadsen, Technical University of Denmark) Jens P. Träff. *LOJel: Learning Objects for Jeliot*, 2011.
181. M. Ben-Ari. *LearnSAT: A DPLL/CDCL/NCP SAT Solver in Prolog*, 2012–17.

Edited Journals

182. M. Ben-Ari (ed.). Special issue on Computer Science Education in High Schools. *Computer Science Education*, 14(1), 2004.
183. M. Ben-Ari, D. Garcia, T. Murphy (eds.). Special Issue on Concurrent, Parallel and Distributed Computation *ACM Transactions on Computing Education*, 2013.

Doctoral student articles supervised

184. Y. Ben-David Kolikant. Gardeners and cinema tickets: high school students' pre-conceptions of concurrency. *Computer Science Education*, 11(3), 2001, 221–245.
185. C. Yehezkel, W. Yurcik, M. Pearson, D. Armstrong. Three simulator tools for teaching computer architecture: EasyCPU, Little Man Computer, and RTLsim. *ACM Journal of Educational Resources in Computing* 2(1), 2002, 60–80.
186. Y. Ben-David Kolikant, S. Pollack. Establishing computer science professional norms among high-school students. *Computer Science Education*, 14(1), 2004, 21-35.
187. Y. Ben-David Kolikant. Learning concurrency: Evolution of students' understanding of synchronization. *International Journal of Human Computers Studies*, 60, 2004, 259-284.
188. Y. Ben-David Kolikant. Learning concurrency as an entry point to the community of computer science practitioners. *Journal of Computers in Mathematics and Science Teaching*, 23(1), 2004, 21-46.
189. C. Yehezkel, W. Yurcik, M. Pearson. Teaching computer architecture with a computer-aided learning environment: State-of-the-art simulators. *International Conference on Simulation and Multimedia in Engineering Education*, 2001, 113–117.
190. C. Yehezkel. A taxonomy of computer architecture visualizations. *Seventh SIGCSE Conference on Innovation and Technology in Computer Science Education*. Aarhus, Denmark, 2002, 101–105.
191. C. Yehezkel. Visualization of computer architecture. *Second Program Visualization Workshop*. HornstrupCentret, Denmark, 2002.
192. C. Yehezkel. Making program execution comprehensible: One level above the machine language. *Eighth SIGCSE Conference on Innovation and Technology in Computer Science Education*. Thessaloniki, Greece, 2003, 124–128.