Mordechai (Moti) Ben-Ari

Professor Emeritus
Department of Science Teaching
Weizmann Institute of Science

Education

- Ph.D. Tel Aviv University, Mathematics (Computer Science), 1981.
- M.Sc. Tel Aviv University, Mathematics (Computer Science), 1977.
- B.Sc., Mathematics, Massachusetts Institute of Technology, 1970.

Awards

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

Postdoctoral Fellows

1. Orni Meerbaum-Salant, 2010-12.
2. Michal Armoni, 2007–2009.

Ph.D. Students

1. Fatima Kaloti-Hallak. *The Effect of Robotics Activities on Students' Learning and Attitudes*, 2015. Joint supervisor: Michal Armoni.
2. Rivka Taub. *The Effect of Computer Science on Physics Learning in a Computational Environment*, 2014. Joint supervisor: Michal Armoni.
3. Jan Lönnberg. *Understanding and Debugging Concurrent Programs through Visualisation*, 2012.
4. Ronit Ben-Bassat Levy. *Teaching Computer Science with Animation: Attitudes and Ways of Experiencing*, 2009.
5. Niko Myller. *Collaborative Software Visualization for Learning: Theory and Applications*, 2009. Joint supervisor: Erkki Sutinen.
6. Noa Ragonis. *Teaching Object-Oriented Programming to Novices*, 2004.
7. Cecile Yehezkel. *A Visualization Environment for Computer Architecture*, 2004. Joint supervisor: Tommy Dreyfus.
8. Yifat Ben-David Kolikant. *Understanding Concurrency: The Process and the Product*, 2003.

M.Sc. Students

1. Mor Friebroon Yesharim. *What do elementary-school students learn in computer science classes with robotics?*, 2018.
2. Fatima Kaloti-Hallak. *Learning Programming Concepts Using Scratch at the Middle-School Level*, 2010. Joint supervisor: Michal Armoni.
3. Rivka Taub. *CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS*, 2010. Joint supervisor: Michal Armoni.
4. Gil Ebel. *The Effect of Program Visualization on the Attention-Directing Characteristics of the Learner*, 2006.

5. Shmuel Schwartz. *Using State Diagrams for Understanding Correctness in Concurrent Programming*, 2006.

6. Andrés Moreno García. *The Design and Implementation of Intermediate Codes for Software Visualization*, 2004. Joint supervisor: Erkki Sutinen.

7. Yevgeniya Kulikova. *Roles of Variables in Teaching Functional Programming*, 2004. Joint supervisor: Jorma Sajaniemi.

8. Maxim Mozgovoy. *Concurrent Program Verifier: A Tool for Teaching Concurrent Programming*, 2004. Joint supervisor: Erkki Sutinen.

9. Niko Myller. *The Fundamental Design Issues of Jeliot 3*. Department of Computer Science, 2004, Joint supervisor: Erkki Sutinen.

10. Tzippora Yeshno. *Teaching an Explicit Conceptual Model As a Means to Improve the Work with Computer Applications*, 2003.

11. Ronit Ben-Bassat Levy. *The Use of Animation as an Educational Tool*, 2002.

12. Yekaterina Sedletzky. *Formal Verification of Distributed Algorithms*, 2000. Joint supervisor: Amir Pnueli.

13. Yakov Persky. *Re-engineering a Concurrency Simulator*, 1999.

14. Yoav Tsruya. *A Distributed Programming Environment in Ada95/Java*, 1998.

15. Noa Ragonis. *Introduction to Expert Systems: Development and Evaluation of a Computer Science Curriculum*, 1997. Joint supervisors: Ehud Shapiro, Zehava Scherz.

16. Ophira Statman. *A Prolog Proof Checker for Temporal Logic*, 1986.

List of Publications

Textbooks

1. M. Ben-Ari. *Principles of Concurrent Programming*. Prentice-Hall International, 1982.

2. M. Ben-Ari. *Principles of Concurrent and Distributed Programming*. Prentice-Hall International, 1990. Second Edition, Addison-Wesley, 2006.

3. M. Ben-Ari. *Mathematical Logic for Computer Science*. Prentice-Hall International, 1992. Second Edition, Springer, 2001. Third Edition, Springer, 2012.

4. M. Ben-Ari. *Understanding Programming Languages*. John Wiley, 1995.

5. M. Ben-Ari. *Ada for Software Engineers.* John Wiley & Sons, 1998. Second Edition with Ada 2005, Springer, 2009.

6. M. Ben-Ari. *Just a Theory: Exploring the Nature of Science*. Prometheus, 2005.

7. M. Ben-Ari. *Principles of the Spin Model Checker*, Springer, 2008.

8. M. Armoni, M. Ben-Ari. *Computer Science Concepts in Scratch*, Weizmann Institute of Science, 2010.

9. M. Ben-Ari, F. Mondada. *Elements of Robotics*, Springer, 2018.
`https://link.springer.com/book/10.1007/978-3-319-62533-1`.

10. M. Ben-Ari. *Mathematical Surprises*, Springer, 2022.
`https://link.springer.com/book/10.1007/978-3-031-13566-8`.

Refereed Journals

11. M. Ben-Ari. Ianov pushdown schemes are contained in boolean recursive schemes. *Acta Informatica* 10(1977), 117–125.

12. M. Ben-Ari. On transposing large $2^n \times 2^n$ matrices. *IEEE Transactions on Computers* C28 (1979), 72–75.

13. M. Ben-Ari. Why you should not time-share. *Software—Practice and Experience* 9(1979), 339–340.

14. M. Ben-Ari. A simplified proof that regular resolution is exponential. *Information Processing Letters* 10(1980), 96–98.

15. M. Ben-Ari. Comments on "Tautology testing with a generalized matrix reduction method". *Theoretical Computer Science* 11(1980), 341.

16. M. Ben-Ari. Cheap concurrent programming. *Software—Practice and Experience* 11(1981), 1261–1264.

17. 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.

18. M. Ben-Ari, Z. Manna, A. Pnueli. The temporal logic of branching time. *Acta Informatica* 20, 1983, 207–226.

19. M. Ben-Ari. Algorithms for on-the-fly garbage collection. *ACM Transactions on Programming Languages and Systems* 6(3), 1984, 333–344.

20. M. Ben-Ari. How to solve the Santa Claus problem. *Concurrency: Practice and Experience* 10(6), 1998, 485–496.

21. M. Ben-Ari. Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching* 20(1), 2001, 45–73.

22. M. Ben-Ari. Interactive execution of distributed algorithms. *ACM Journal on Educational Resources in Computing*, 1(2), 2001.

23. M. Ben-Ari. Theory-guided technology in computer science. *Science & Education* 10(5), 2001, 477–484.

24. R. Ben-Bassat Levy, M. Ben-Ari, P.A. Uronen. The Jeliot 2000 program animation system. *Computers & Education* 40(1), 2003, 1–15.

25. M. Ben-Ari. The NOMA of Yishayahu Leibowitz. *Science & Education* 12(7), 2003, 719–723.

26. M. Ben-Ari. On random numbers and design. *Science & Education* 13(3), 2004, 235–241.

27. M. Ben-Ari. Situated learning in computer science education. *Computer Science Education* 14(2), 2004, 85–100.

28. M. Ben-Ari. Situated learning in this high-technology world. *Science & Education* 14(3-5), 2005, 367–376.

29. 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.

30. M. Ben-Ari, T. Yeshno. Conceptual models of software artifacts. *Interacting with Computers* 18 (6), 2006, 1336–1350.

31. 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.

32. C. Yehezkel, M. Ben-Ari, T. Dreyfus. The contribution of visualization to learning computer architecture. *Computer Science Education* 17(2), 2007, 117–127.

33. Y. Ben-David Kolikant, M. Ben-Ari. Fertile zones of cultural encounter. *Journal of the Learning Sciences* 17(1), 2008, 1–32.

34. R. Ben-Bassat Levy, M. Ben-Ari. Adapting and merging methodologies in doctoral research. *Computer Science Education* 19(2), 2009, 51–67.

35. 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.

36. 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.

37. 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.

38. 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.

39. O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 2013, 239–264.

40. M. Armoni, O. Meerbaum-Salant, M. Ben-Ari. From Scratch to "Real" Programming. *ACM Transactions on Computing Education*, 14(4), article 25, 2015.

41. 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.

42. 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.

43. 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`.

44. M. Friebroon-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.

45. R. Taub, M. Ben-Ari, M. Armoni. Physics conceptual understanding in a computational science course. *Journal of Computational Science Education* 9(2), 2018.

46. 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

47. 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.

48. 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.

49. 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.

50. M. Ben-Ari. Reverse engineering into Ada. *Workshop on Software Methodologies in Ada*, Tel-Aviv, Israel, 1983.

51. 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).

52. M. Ben-Ari. Ada requirements for small real-time systems. *Third International Workshop on Real-Time Ada Issues*. Nemacolin Woodlands, PA, 1989, 159–165.

53. M. Ben-Ari. Experience teaching object-oriented programming in Ada. *Symposium on Teaching Object Technology*. Santa Barbara, CA, 1996.

54. M. Ben-Ari. Using inheritance to implement concurrency. *Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education*. Philadelphia, PA, 1996, 180–184.

55. M. Ben-Ari. Distributed algorithms in Java. *Second SIGCSE Conference on Integrating Technology into Computer Science Education*. Uppsala, Sweden, 1997, 62-64.

56. M. Ben-Ari. The software factory. *Psychology of Programming Interest Group Tenth Annual Workshop*. Milton Keynes, UK, 1998, 89–91.

57. M. Ben-Ari. Constructivism in computer science education. *Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education*. Atlanta, GA, 1998, 257–261.

58. 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.

59. 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.

60. 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.

61. M. Ben-Ari. Bricolage forever! *Eleventh Annual Workshop of the Psychology of Programming Interest Group*. Leeds, UK, 1999, 53–57.

62. 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.

63. 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.

64. M. Ben-Ari. Theory-guided technology in computer science. *Fifth International History, Philosophy and Science Teaching Conference*. Como, Italy, 1999.

65. 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.

66. 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.

67. 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.

68. T. Yeshno, M. Ben-Ari. Salvation for bricoluers. *Thirteenth Annual Workshop of the Psychology of Programming Interest Group*. Bournemout, UK, 2001, 225–235.

69. M. Ben-Ari. The argument *for* design. *Sixth International History, Philosophy and Science Teaching Conference*. Denver, CO, 2001.

70. 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.

71. 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.

72. M. Ben-Ari, N. Ragonis, R. Ben-Bassat Levy. A vision of visualization in teaching object-oriented programming. *Second Program Visualization Workshop*. HornstrupCentret, Denmark, 2002, 83–89.

73. 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.

74. M. Ben-Ari. Situated learning in this high-technology world. *Seventh International History, Philosophy and Science Teaching Conference*. Winnipeg, Canada, 2003.

75. 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.

76. A. Moreno, N. Myller, E. Sutinen, M. Ben-Ari. Visualizing programs with Jeliot 3. *Conference on Advanced Visual Interfaces*, Gallipoli, Italy, 2004, 373–376.

77. 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.

78. S. Pollack, M. Ben-Ari. Selecting a visualization system. *Third Program Visualization Workshop*. Warwick, UK, 2004, 121–126.

79. C. Yehezkel, M. Ben-Ari, T. Dreyfus. Inside the computer: Visualization and mental models. *Third Program Visualization Workshop*. Warwick, UK, 2004, 77–80.

80. 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.

81. 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.

82. M. Ben-Ari. Whose final hour? The rise of naive egocentric catastrophism. *Eighth International History, Philosophy and Science Teaching Conference*. Leeds, UK, 2005.

83. 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.

84. G. Ebel, M. Ben-Ari. Affective effects of program visualization. *Second International Computing Education Research Conference*. Canterbury, UK, 2006, 1–5.

85. 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.

86. 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.

87. 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.

88. 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.

89. 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.

90. 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.

91. M. Ben-Ari. Teaching concurrency and model checking. *16th International SPIN Workshop on Model Checking of Software*, Grenoble, France, 2009, 6–11.

92. 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.

93. 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.

94. J. Lönnberg, M. Ben-Ari, L. Malmi. Java replay for dependence-based debugging. *Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging (PADTAD-IX)*, Toronto, ON, 2011, 15–25.

95. 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.

96. 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.

97. 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.

98. M. Ben-Ari. LearnSAT: A SAT solver for education. *16th Int. Conf. on Theory and Applications of Satisfiability Testing*, Helsinki, Finland, 2013, 403–407.

99. 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.

100. 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.

101. 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.

102. 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.

103. 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.

104. 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.

105. 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.

Pedagogical Software Tools

Download from GitHub at `https://github.com/motib`, except for the Scratch projects which are at `https://scratch.mit.edu/users/MotiB`.

106. M. Ben-Ari. *CP: A Concurrency Simulator*, 1974–80, 1996–2000.

107. M. Ben-Ari. *DAJ: Interactive Learning of Distributed Algorithms*, 2001, 2005.

108. (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.

109. M. Ben-Ari. *jBACI: Development Environment and Simulator for Concurrency*, 2000–04.

110. M. Ben-Ari. *jSpin: Development Environment for the Spin Model Checker*, 2004–07.

111. M. Ben-Ari. *VN: Visualizing Nondeterminism*, 2006.

112. M. Ben-Ari. *The Erigone Model Checker*, 2009–11.

113. M. Ben-Ari. *Scratch projects: Karel the Robot, Valentino the Robot, Computer Science Unplugged*.

114. M. Ben-Ari. *LearnSAT: A DPLL/CDCL/NCP SAT Solver in Prolog*, 2012–17.