

Publication List: Shmuel Katz

- Journal Articles

1. S. Katz and Z. Manna, A closer look at termination, *Acta Informatica*, 5, December 1975, pp. 333-352.
2. S. Katz and Z. Manna, Logical analysis of programs, *CACM*, 19, 4, April 1976, pp. 188-206.
3. S. Katz, Program optimization using invariants, *IEEE Transactions on Software Engineering*, vol. SE-4, no. 5, September 1978, pp. 378-389.
4. S. Katz, L. Risman and M. Rodeh, A system for constructing linear programming models, *IBM Systems Journal*, vol. 19, no. 4, November 1980, pp. 505-520.
5. N. Francez, O. Grumberg, and S. Katz, A complete rule for equifair termination, *Journal of Computer and System Sciences*, vol. 33, December 1986, pp. 312-332.
6. S. Katz and O. Shmueli, A cooperative distributed algorithm for dynamic cycle prevention, *IEEE Transactions on Software Engineering*, vol. SE-13, no. 5, May 1987, pp. 540-552.
7. N. Francez and S. Katz, Fairness and the axioms of control predicates, *International Journal of Parallel Programming*, vol. 16, no. 4, August 1987, pp. 263-278.
8. O. Baruch and S. Katz, Partially interpreted schemas for CSP programming, *The Science of Computer Programming*, vol. 10, no. 1, February 1988, pp. 1-18.
9. K. Apt, N. Francez, and S. Katz, Appraising fairness in languages for distributed programming, *Distributed Computing*, vol. 2, no. 4, August 1988, pp. 226-241.
10. M. Evangelist, N. Francez, and S. Katz, Multiparty interactions for interprocess communication and synchronization, *IEEE Transactions on Software Engineering*, vol. 15, no. 11, November 1989, pp. 1417-1426.
11. G. Taubenfeld, S. Katz, and S. Moran, Initial failures in distributed computations, *International Journal of Parallel Programming*, vol. 18, no. 4, August 1989, pp. 255-276.
12. S. Katz, C. Richter, and K.C. The, PARIS: a system for reusing partially interpreted schemas, in *Software Reusability, Volume I: Concepts and Models*, T. Biggerstaff and A.J. Perlis, eds., ACM Press (Addison-Wesley), 1989, pp. 257-273. (Note: refereed chapter in collection of articles)
13. S. Katz and D. Peled, Interleaving set temporal logic, *Theoretical Computer Science*, vol. 75, October 1990, pp. 263-287.

14. G. Goldszmidt, S. Katz, and S. Yemini, High level language debugging for concurrent programs, *ACM Transactions on Computer Systems*, vol. 8, no. 4, November 1990, pp. 311-336.
15. S. Katz and D. Peled, Defining conditional independence using collapses, *Theoretical Computer Science*, vol. 101, no. 2, July 1992, pp. 337-359.
16. S. Katz and D. Peled, Verification of distributed programs using representative interleaving sequences, *Distributed Computing*, vol. 6, no. 2, September 1992, pp. 107-120.
17. S. Katz, A superimposition control construct for distributed systems, *ACM Trans. on Programming Languages and Systems*, vol. 15, no. 2, April 1993, pp. 337-356.
18. S. Katz and K. Perry, Self-stabilizing extensions for message-passing systems, *Distributed Computing*, vol. 7, no. 1, 1993, pp. 17-26.
19. G. Taubenfeld, S. Katz, and S. Moran, Impossibility results in the presence of multiple faulty processes, *Information and Computation*, vol. 113, no. 2, September 1994, pp. 173-198.
20. S. Katz, Refinement with global equivalence proofs in temporal logic, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 29, 1997, pp. 59-78.
21. S. Katz and H. Miller, Saving space by fully exploiting invisible transitions, *Formal Methods in System Design*, vol. 14, no. 3, 1999, pp. 311-332.
22. Z. Collin, R. Dechter, and S. Katz, Self-stabilizing distributed constraint satisfaction, *Chicago Journal of Theoretical Computer Science*, vol. 1999, no. 10, December 1999.
23. I. Reinhartz-Berger, D. Dori, and S. Katz, OPM/Web—Object-Process Methodology for developing Web applications, *The Annals of Software Engineering*, vol. 13, 2002, pp. 141-161.
24. M. Glusman and S. Katz, A mechanized proof environment for the convenient computations proof method, *Formal Methods in System Design*, vol. 23, no. 2, 2003, pp. 115-142.
25. M. Sihman and S. Katz, Superimpositions and aspect-oriented programming, *BCS Computer Journal*, vol. 46, no. 5, 2003, pp. 529-541.
26. I. Reinhartz-Berger, D. Dori, and S. Katz, Modelling code mobility and migration: An OPM/Web Approach, *International Journal of Web Engineering and Technology (IJWET)*, vol. 2, no. 1, 2005, pp. 6-28.
27. S. Katz, Aspect categories and classes of temporal properties, *Transactions on Aspect-Oriented Software Development (TAOSD)*, vol. 1, LNCS 3880, 2006, pp. 106-134.
28. J. Brichau, R. Chitchyan, S. Clarke, E. D'Hondt, A. Garcia, M. Haupt, W. Joosen, S. Katz, J. Noyé, A. Rashid, and M. Südholt, A Model Curriculum for Aspect-Oriented Software Development, *IEEE Software*, vol. 23, no. 6, 2006, pp. 53-61.
29. O. Grumberg and S. Katz, VeriTech—a framework for translating among model description notations, *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 9, no. 2 (DOI 10.1007/s10009-006-0003-0), 2007, pp. 119-132.

30. M. Katara and S. Katz, A concern architecture view for aspect-oriented software design, *Software and Systems Modeling (SoSyM)*, vol. 6, 2007 (DOI 10.1007/s10270-006-0032-x), pp. 247-265.
 31. I. Reinhartz-Berger, D. Dori, and S. Katz, Reusing semi-specified behavior models in systems analysis and design, *Software and Systems Modeling (SoSyM)*, vol. 8, 2009 DOI 10.1007/s10270-007-0079-3, pp. 221-234.
 32. M Goldman, E. Katz, and S. Katz, Maven: modular aspect verification and interference analysis, *Formal Methods in System Design*, vol. 37, 2010. Springer Online first, DOI 10.1007/s10703-010-0101-1.
- Surveys and Invited Chapters in Books
 1. S. Katz and Z. Manna, A closer look at termination, in *Current Trends in Programming Methodology: Vol. II, Program Validation*, R. Yeh, ed., Prentice-Hall, 1977, pp. 248-268.
 2. S. Katz and Z. Manna, Logical analysis of programs, in *Studies in Automatic Programming Logic*, edited by Z. Manna and R. Waldinger, North-Holland, 1978, pp. 93-140.
 3. S. Katz, Exploiting independence for verification, refinement, and modularity, in *Models, Algebras, and Logic of Engineering Software*, M. Broy and M. Pizka, eds., IOS Press, 2003, pp. 195-208.
 4. S. Katz, On translating models and properties, in *Models, Algebras, and Logic of Engineering Software*, M. Broy and M. Pizka, eds., IOS Press, 2003, pp. 209-226.
 - Articles in Refereed Conference Proceedings
 1. S. Katz and Z. Manna, A heuristic approach to program verification, Proceedings of Third International Conference on Artificial Intelligence, Stanford, August 1973, pp. 500-512.
 2. S. Katz and Z. Manna, Towards automatic debugging of programs, Proceedings of International Conference on Reliable Software, *Sigplan Notices*, June 1975, pp. 143-155.
 3. S. Katz, Program optimization using invariants, Proceedings of Conference of the ECI, *Lecture Notes in Computer Science*, Vol. 44, Springer-Verlag, K. Samuelson, ed., August 1976, pp. 287-309,
 4. S. Rosenschein and S. Katz, Selection of representations for data structures, Proceedings of a Symposium on Artificial Intelligence and Programming Languages, *Sigplan Notices*, Vol. 12, No. 8, August 1977, pp. 147-154.
 5. S. Katz and R. Zimmerman, An advisory system for developing data representations, Proceedings of IJCAI-81, Vancouver, Canada, August 1981, pp. 1030-1036.
 6. N. Francez, O. Grumberg and S. Katz, A complete proof rule for equifair termination, Proceedings of Second Workshop on Logics of Programs, Carnegie-Mellon University, *Lecture Notes in Computer Science*, no. 164 (E. Clarke, D. Kozen, eds.), Springer-Verlag, 1983, pp. 257-278.

7. N. Francez, O. Grumberg, and S. Katz, Fair termination of communicating processes, Proceedings of Third Annual ACM Symposium on Principles of Distributed Computing, Vancouver, Canada, August 1984, pp. 254-265.
8. N. Francez, O. Grumberg, S. Katz and A. Pnueli, Proving termination of PROLOG programs, Proceedings of Third Workshop on Logics of Programs, Brooklyn, New York, *Lecture Notes in Computer Science*, no. 193 (R. Parikh, ed.), Springer-Verlag, 1985, pp. 89-105.
9. S. Katz and G. Taubenfeld, What processes know: definitions and proof methods, Proceedings of Fifth Annual ACM Symposium on Principles of Distributed Computing, Calgary, Canada, August 1986, pp. 249-262.
10. K. Apt, N. Francez, and S. Katz, Appraising fairness in languages for distributed programming, Proceedings of the 14th Annual ACM Symposium on Principles of Programming Languages, Munich, West Germany, January 1987, pp. 189-198.
11. S. Katz, C. Richter, and K.S. The, PARIS: a system for reusing partially interpreted schemas, Proceedings of the Ninth International Conference on Software Engineering, Monterey, California, April 1987, pp. 377-385.
12. S. Katz and D. Peled, Interleaving set temporal logic, Proceedings of Sixth Annual ACM Symposium on Principles of Distributed Computing, Vancouver, Canada, August, 1987, pp. 178-190.
13. S. Katz and D. Peled, An efficient verification method for parallel and distributed programs, Proceedings of Workshop on Linear Time, Branching Time and Partial Orders in Logics and Models for Concurrency, *Lecture Notes in Computer Science*, no. 354 (J.W. de Bakker, W.P. de Roever, G. Rozenburg, eds.), Springer-Verlag, June 1988, pp. 489-507.
14. S. Katz, I. Forman, and M. Evangelist, Language constructs for distributed systems, in Programming Concepts and Methods (Proceedings of IFIP TC2 Working Conference, Sea of Galilee, Israel, April, 1990), M. Broy and C.B. Jones, eds., North-Holland, pp. 73-100.
15. S. Katz and D. Peled, Defining conditional independence using collapses, Proceedings of International Workshop on Semantics of Concurrency, Leicester, England, Springer-Verlag, July 1990, pp. 262-280.
16. S. Katz and K. Perry, Self-stabilizing extensions for message-passing systems, Proceedings of 9th ACM Symposium on Principles of Distributed Computing, Quebec, Canada, August 1990, pp. 91-102.
17. D. Peled, S. Katz, and A. Pnueli, Specifying and proving serializability in temporal logic, Proceedings of 6th IEEE Symposium on Logic in Computer Science, Amsterdam, Holland, July 1991, pp. 232-244.
18. Z. Collin, R. Dechter, and S. Katz, On the feasibility of distributed constraint satisfaction, Proceedings of 12th International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney, Australia, August 1991, pp. 318-324.
19. S. Katz, Global equivalence proofs in ISTL, Proceedings of International Conference on Temporal Logic, *Lecture Notes in Artificial Intelligence*, no. 827, Springer-Verlag, Bonn, Germany, July 1994, pp. 17-29.

20. J. Howard and S. Katz, Reconciliations, Proceedings of Thirteenth ACM Symposium on Principles of Distributed Computing, Los Angeles, August 1994, pp. 14-21.
21. A. Adir, N. Francez, and S. Katz, Implementing a language for coordinated distributed computing on a transputer multiprocessor, Proceedings of World Transputer Congress, IOS Press, Como, Italy, 1995, pp. 446-463.
22. H. Miller and S. Katz, Saving space by fully exploiting invisible transitions, Proceedings of CAV-96 (Computer-Aided Verification), LNCS 1102, July 1996, pp. 336-347.
23. S. Katz, P. Lincoln, and J. Rushby, Low-overhead time-triggered group membership, Proceedings of Workshop on Distributed Algorithms (WDAG), LNCS 1320, September 1997, pp. 155-169.
24. M. Glusman and S. Katz, Mechanizing proofs of computation equivalence, Proceedings of CAV-99 (Computer-Aided Verification), LNCS 1633, 1999, pp. 354-368.
25. S. Katz, Faithful translations among models and specifications, Proceedings of FME-2001 (Formal methods–Europe), LNCS 2021, 2001, pp. 419-434.
26. M. Glusman and S. Katz, Extending memory consistency of finite prefixes to infinite computations, CONCUR-2001 (International Conference on Concurrency Theory), LNCS 2154, 2001, pp. 411-425.
27. M. Sihman and S. Katz, A calculus of superimpositions for distributed systems, Proceedings of AOSD-2002 (International Conference on Aspect-Oriented Software Development), ACM Press, April, 2002, pp. 28-41.
28. K. Korenblat, O. Grumberg, and S. Katz, Translations between textual transition systems and Petri nets, Proceedings of IFM2002 (International Conference on Integrated Formal Methods), LNCS 2335, May 2002, pp. 339-359.
29. S. Katz and O. Grumberg, A framework for translating models and specifications, Proceedings of IFM2002 (International Conference on Integrated Formal Methods), LNCS 2335, May 2002, pp. 145-164.
30. I. Reinhartz-Berger, D. Dori, and S. Katz, Open reuse of component designs in OPM/Web, Proceedings of Compsac2002, IEEE Computer Society, August 2002, pp. 19-24.
31. M. Katara and S. Katz, Architectural views of aspects, Proceedings of AOSD-2003 (International Conference on Aspect-Oriented Software Development), ACM Press, March 2003, pp. 1-10.
32. M. Sihman and S. Katz, Model checking applications of aspects and superimpositions, FOAL Workshop (Foundations of Aspect-Oriented Languages, associated with AOSD2003), March 2003, pp. 51-60.
33. M. Glusman and S. Katz, Model checking conformance with scenario-based specifications, CAV03 (International Conference on Computer-Aided Verification), LNCS 2725, July 2003, pp. 328-340.
34. S. Katz, Diagnosis of harmful aspects using regression verification, FOAL Workshop, associated with AOSD2004, March 2004, pp. 1-6.
35. S. Katz and A. Rashid, From aspectual requirements to proof obligations for aspect-oriented systems, RE04 (International Requirements Engineering Conference), IEEE Computer Society, September 2004, pp. 48-57.

36. E. Katz and S. Katz, Verifying scenario-based aspect specifications, FM05 (Formal Methods 2005), LNCS 3582, July 2005, pp. 432-447.
37. O. Mishali and S. Katz, Using aspects to support the software process: XP over Eclipse, AOSD2006 (International Conference on Aspect-Oriented Software Development), ACM Press, March 2006, pp. 169-179.
38. M. Goldman and S. Katz, Modular generic verification of LTL properties for aspects, FOAL Workshop, associated with AOSD2006, March 2006.
39. S. Katz, Rigorous fault tolerance using aspects and formal methods, Rigorous Development of Complex Fault-Tolerant Systems (refereed postconference proceedings), LNCS 4157, November 2006, pp. 226-240.
40. M. Goldman and S. Katz, MAVEN: Modular Aspect Verification, TACAS07 (Tools and Algorithms for the Construction and Analysis of Systems), LNCS 4424, March 2007, pp. 308-322.
41. E. Katz and S. Katz, Iterative analysis of interference among aspects, FOAL Workshop, associated with AOSD2008, April 2008.
42. O. Mishali, Y. Dubinsky, and S. Katz, The TTD-Guide training and guidance tool for test-driven development, XP08 (Intl. Conf. on Agile Processes and Extreme Programming in Software Engineering), June 2008.
43. E. Katz and S. Katz, Modular Verification of Strongly Invasive Aspects: Summary, FOAL Workshop, associated with AOSD2009, ACM Digital Library, 2009.
44. O. Mishali and S. Katz, The HighspectJ Framework, ACP4IS workshop, associated with AOSD2009, ACM Digital Library, 2009.
45. E. Katz and S. Katz, Modular Verification of Strongly Invasive Aspects, Symposium on Languages: from Formal to Natural, LNCS 5533, May 2009, pp. 128-147.
46. E. Katz and S. Katz, Semantic aspect interactions and possibly shared join points, FOAL workshop, associated with AOSD 2010.
47. E. Katz and S. Katz, User queries for specification refinement treating shared aspect join-points, 8th Intl. Conf. on Software Engineering and Formal Methods (SEFM), 2010.
48. S. Cirici, S. Malakuti, S. Katz, and M. Aksit, Checking the correspondence between UML models and implementation, Intl. Conf. on Runtime Verification (RV), LNCS 6418, pp. 198-213, Sept. 2010.