Classified Larch Bibliography

Stephen J. Garland

October 22, 1996

This classified bibliography is an updated version of Appendix C in [44]. Papers of historical interest only are included among the references, but do not appear in the classification.
The LATEXfiles used to prepared this bibliography are available over the WWW from the Larch home page, `http://larch.lcs.mit.edu:8001/larch/index.html`. `ftp` from Internet node `larch.lcs.mit.edu` as
Please send corrections and suggested additions to `larch@lcs.mit.edu`. Kindly include a brief description of each suggested addition so that it can be cited in appropriate categories.

Papers about Larch

Current work

An overview of Larch is available in [44], which describes the Larch Shared Language and the Larch/C interface language. It also provides an introduction to the Larch Prover and the Larch/Modula-3 interface language.
Reports about the current status of several Larch-related projects are contained in [61].

Larch languages

Larch Shared Language: [44, 50]
Larch Interface Languages

- Generic [11, 48, 56, 86]
- Ada [31]
- C [20, 44, 72, 74, 73, 75]
- C++ [55]
- CLU [84]
- ML [92]
- Modula-3 [51, 52, 53, 44]
- Smalltalk [12]

Larch and other methods: [2, 44, 36, 58, 94]

Larch tools

LP, the Larch Proof Assistant

- User's Guide: [24, 25]
- Distributed LP (DLP): [81]

- Beginner's strategy guide: [76]
- Extensions: [82]
- Use: [4, 7, 9, 13, 14, 68, 83]

For LSL: [6, 54]
For LCL: [17, 19, 18, 20]
For LM3: [51]

Sample specifications

Apple MAC Toolbox: [8]
Avalon built-in classes, examples (queue, directory, counter): [56, 87, 88, 91]
Database: [44]
Display: [38]
Distributed algorithms: [59, 64, 69]
Finite element analysis library: [1, 3]
Garbage collection: [15]
IOStreams: [52]
Larch/Ada: [10, 31]
Library: [85]
Miro languages and editor: [93, 95]
Thread synchronization primitives: [5, 63]
Using specifications to search software libraries: [65]

Proofs using LP

Ada programs: [32]
Analysis of Larch specifications: [44]
Avalon queue example: [30, 90, 91]
Circuit examples: [13, 28, 62, 67, 70, 71]
Compiler optimizations: [77, 78, 79, 80].
Distributed algorithms: [59, 64, 69]
Mathematical Theorems: [60]
Temporal Logic of Actions: [16]

References

[1] John W. Baugh, Jr. Formal specification of engineering analysis programs. In E. N. Houstis, J. R Rice, and R. Vichnevetsky, editors, *Expert Systems for Numerical Computing*. North-Holland, 1992.

[2] John W. Baugh, Jr. Is engineering software amenable to formal specification? In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[3] John W. Baugh, Jr. and D. R. Rehak. Computational abstractions for finite element programming. TR 89–182, Department of Civil Engineering, Carnegie Mellon University, September 1989.

[4] Michel Bidoit and Rolf Hennicker. How to prove observational theorems with LP. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[5] Andrew D. Birrell, John V. Guttag, James J. Horning, and Roy Levin. Synchronization primitives for a multiprocessor: A formal specification. *Operating Systems Review*, 21(5), November 1987. Also published as DEC Systems Research Center Report 20, August 1987. Revised version in [63].

[6] Robert H. Bourdeau and Betty H. C. Cheng. An object-oriented toolkit for constructing specification editors. In *COMPSAC'92: Computer Software and Applications Conference*, September 1992.

[7] Manfred Broy. Experiences with software specification and verification using LP, the Larch Proof Assistant. Report 93, DEC Systems Research Center, Palo Alto, CA, October 1992.

[8] C. T. Burton, S. J. Cook, S. Gikas, J. R. Rowson, and S. T. Sommerville. Specifying the Apple Macintosh toolbox event manager. *Formal Aspects of Computing*, 1(2):147–171, 1989.

[9] Karl-Heinz Buth. Using SOS definitions in term-rewriting proofs. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[10] S. R. Cardenas and H. Oktaba. Formal specification in Larch case study: Text manager. Interface specification, implementation, in Ada and validation of implementation. TR 511, Instituto de Investigaciones en Matematicas Aplicadas y en Sistemas, Universidad Nacional Autonoma de Mexico, 1988.

[11] Jolly Chen. The Larch/Generic interface language. S. B. Thesis, Department of Electrical Engineering and Computer Science, MIT, 1989.

[12] Yoonsik Cheon. Larch/Smalltalk: A specification language for Smalltalk. Master's thesis, Iowa State University, 1991.

[13] Boutheina Chetali and Pierre Lescanne. An exercise in LP: the proof of a nonrestoring division circuit. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[14] Christine Choppy and Michel Bidoit. Integrating ASSPEGIQUE and LP. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[15] David L. Detlefs. Concurrent, atomic garbage collection. TR CS–90–177, Carnegie Mellon University, October 1990. Ph. D. Thesis, Department of Computer Science.

[16] Urban Engberg, Peter Grønning, and Leslie Lamport. Mechanical verification of concurrent systems with TLA. In *Workshop on Computer Aided Verification*, 1992. Revised version in [61].

[17] David Evans. Using specifications to check source code. Technical Report MIT/LCS/TR–628, MIT Department of Electrical Engineering and Computer Science, June 1994. SM Thesis.

[18] David Evans. Static detection of dynamic memory errors. In *SIGPLAN Conference on Programming Language Design and Implementation, PLDI'96*, Philadelphia, PA, May 1996.

[19] David Evans, John V. Guttag, James J. Horning, and Yang Meng Tan. LCLint: A tool for using specifications to check code. In *Symposium on the Foundations of Software Engineering*, December 1994.

[20] Gary Feldman and Joseph Wild. The DECspec project: Tools for Larch/C. In *Fifth International Workshop on Computer-Aided Software Engineering*, Montreal, July 1992. Revised version in [61].

[21] Stephen J. Garland and John V. Guttag. Inductive methods for reasoning about abstract data types. In *15th ACM Symposium on Principles of Programming Languages*, pages 219–228, San Diego, January 1988.

[22] Stephen J. Garland and John V. Guttag. An overview of LP, the Larch Prover. In *Third International Conference on Rewriting Techniques and Applications*, pages 137–151, Chapel Hill, 1989. Springer-Verlag Lecture Notes in Computer Science 355.

[23] Stephen J. Garland and John V. Guttag. Using LP to debug specifications. In M. Broy and C. B. Jones, editors, *Programming Concepts and Methods*, pages 369–386, Sea of Galilee, Israel, April 2–5 1990. IFIP WG2.2/2.3, North-Holland. Superseded by [26].

[24] Stephen J. Garland and John V. Guttag. A guide to LP, the Larch Prover. Report 82, DEC Systems Research Center, Palo Alto, CA, December 1991.

[25] Stephen J. Garland and John V. Guttag. *LP, the Larch Prover: Version 3.1.* MIT Laboratory for Computer Science, January 1995. Available as http://larch.lcs.mit.edu:8001/larch/LP/overview.html.

[26] Stephen J. Garland, John V. Guttag, and James J. Horning. Debugging Larch Shared Language specifications. *IEEE Transactions on Software Engineering*, 16(9):1044–1075, September 1990. Reprinted as DEC Systems Research Center Report 60. Superseded by Chapter 7 in [44].

[27] Stephen J. Garland, John V. Guttag, and James J. Horning. An overview of Larch. In Peter E. Lauer, editor, *Functional Programming, Concurrency, Simulation and Automated Reasoning*, pages 329–348. Springer-Verlag Lecture Notes in Computer Science 693, 1993.

[28] Stephen J. Garland, John V. Guttag, and Jørgen A. Staunstrup. Verification of VLSI circuits using LP. In *The Fusion of Hardware Design and Verification*, pages 329–345, Glasgow, July 4–6 1988. IFIP WG 10.2, North Holland.

[29] Narain Gehani and Andrew McGettrick, editors. *Software Specification Techniques.* Addison-Wesley, 1986.

[30] Chun Gong and Jeannette M. Wing. Raw code, specification, and proofs of the Avalon queue example. TR CMU–CS–89-172, Carngie Mellon University, August 1989.

[31] David Guaspari, Carla Marceau, and Wolfgang Polak. Formal verification of Ada. *IEEE Transactions on Software Engineering*, 16(9):1058–1075, September 1990.

[32] David Guaspari, Carla Marceau, and Wolfgang Polak. Formal verification of Ada programs. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch.* Springer-Verlag, July 1992.

[33] John V. Guttag. Dyadic specification and its impact on reliability. In J. E. Donahue, J. D. Gannon, Guttag J. V., and J. J. Horning, editors, *Three Approaches to Reliable Software: Language Design Dyadic Specification, Complementary Semantics.* University of Toronto, December 1974. TR CSRG–45.

[34] John V. Guttag. *The Specification and Application to Programming of Abstract Data Types.* PhD thesis, University of Toronto, Department of Computer Science, 1975.

[35] John V. Guttag. Notes on type abstraction. In *Conference on Specifications of Reliable Software*, 1979. Reprinted in [29].

[36] John V. Guttag. Goldilocks and the three specifications. In *Proceedings of TAPSOFT '93.* Springer-Verlag, April 1993.

[37] John V. Guttag and James J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10(1), 1978.

[38] John V. Guttag and James J. Horning. Formal specification as a design tool. In *7th ACM Symposium on Principles of Programming Languages*, pages 251–261, Las Vegas, January 1980. Reprinted in [29].

[39] John V. Guttag and James J. Horning. An introduction to the Larch Shared Language. In *IFIP Ninth World Computer Conference*, pages 809–814, Paris, September 1986. Superseded by Chapter 3 of [44].

[40] John V. Guttag and James J. Horning. A Larch Shared Language handbook. *Science of Computer Programming*, 6(2), March 1986. Superseded by Appendix A of [44].

[41] John V. Guttag and James J. Horning. Report on the Larch Shared Language. *Science of Computer Programming*, 6(2), March 1986. Superseded by [45] and Chapter 4 of [44].

[42] John V. Guttag and James J. Horning. LCL: A Larch interface language for C. Report 74, DEC Systems Research Center, Palo Alto, CA, July 1991. Superseded by Chapter 5 of [44].

[43] John V. Guttag and James J. Horning. A tutorial on Larch and LCL, a Larch/C interface language. In S. Prehn and W. J. Toetenel, editors, *VDM91: Formal Software Development Methods*, Delft, October 1991. Springer-Verlag Lecture Notes in Computer Science 551. Superseded by Chapter 3 of [44].

[44] John V. Guttag and James J. Horning, editors. *Larch: Languages and Tools for Formal Specification*. Texts and Monographs in Computer Science. Springer-Verlag, 1993. With Stephen J. Garland, Kevin D. Jones, Andrés Modet, and Jeannette M. Wing.

[45] John V. Guttag, James J. Horning, and Andrés Modet. Report on the Larch Shared Language: Version 2.3. Report 58, DEC Systems Research Center, Palo Alto, CA, April 14 1990. Superseded by Chapter 4 of [44].

[46] John V. Guttag, James J. Horning, and Jeannette M. Wing. The Larch family of specification languages. *IEEE Software*, 2(5):24–36, 1985. Superseded by Chapter 3 of [44].

[47] John V. Guttag, James J. Horning, and Jeannette M. Wing. Larch in five easy pieces. Report 5, DEC Systems Research Center, Palo Alto, CA, July 1985. Superseded by [44].

[48] David Hinman. On the design of Larch interface languages. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, January 1987.

[49] James J. Horning. Combining algebraic and predicative specifications in Larch. In *International Joint Conference on Theory and Practice of Software Development, TAPSOFT*, pages 12–26, Berlin, March 1985. Springer-Verlag Lecture Notes in Computer Science 186. Superseded by [44].

[50] James J. Horning. The Larch Shared Language: Some open problems. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification*, Berlin, 1996. Springer-Verlag Lecture Notes in Computer Science 1130. 11th Workshop on Specification of Abstract Data Types, Joint with the 8th COMPASS Workshop, Oslo, Norway, September 1995, Selected Papers, ISSN 0302-9743, ISBN 3-540-61629-2.

[51] Kevin D. Jones. LM3 reference manual. In preparation.

[52] Kevin D. Jones. LM3: A Larch interface language for Modula-3: Version 1.0. Report 72, DEC Systems Research Center, Palo Alto, CA, June 1991.

[53] Kevin D. Jones. A semantics for a Larch/Modula-3 interface language. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[54] Michael R. Laux, Robert H. Bourdeau, and Betty H. C. Cheng. An integrated environment supporting the reuse of formal specifications. TR MSU–CPS–ACS–70, Michigan State University, Deptartment of Computer Science, September 1992.

[55] Gary T. Leavens and Yoonsik Cheon. Preliminary design of Larch/C++. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[56] Richard Allen Lerner. *Specifying Objects of Concurrent Systems*. PhD thesis, School of Computer Science, Carnegie Mellon University, May 1991. TR CMU–CS–91–131.

[57] Pierre Lescanne. Computer experiments with the REVE term-rewriting system generator. In *Tenth ACM Symposium on Principles of Programming Languages*, 1983.

[58] Barbara Liskov and John V. Guttag. *Abstraction and Specification in Program Development*. The MIT Electrical Engineering and Computer Science Series. MIT Press, Cambridge, MA, 1986.

[59] Victor Luchangco, Ekrem Söylemez, Stephen J. Garland, and Nancy A. Lynch. Verifying timing properties of concurrent algorithms. In *FORTE'94: Seventh International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols*, pages 259–273, Berne, Switzerland, October 4–7 1994. Chapman and Hall.

[60] Ursula Martin and Tobias Nipkow. Automating Squiggol. In M. Broy and C. B. Jones, editors, *Programming Concepts and Methods*, pages 223–236, Sea of Galilee, Israel, April 2–5 1990. IFIP WG2.2/2.3, North-Holland.

[61] Ursula Martin and Jeannette M. Wing, editors. *First International Workshop on Larch*, Dedham, MA, July 1992. Springer-Verlag.

[62] Niels Mellergaard and Jørgen A. Staunstrup. Generating proof obligations for circuits. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[63] Greg Nelson, editor. *Systems Programming with Modula-3*. Prentice Hall, 1991.

[64] Tsvetomir P. Petrov, A. Pogosyants, , Stephen J. Garland, Victor Luchangco, and Nancy A. Lynch. Computer-assisted verification of an algorithm for concurrent timestamps. In Reinhard Gotzhein and Jan Bredereke, editors, *Formal Description Techniques XI: Theory, Application, and Tools*, pages 29–44, Kaiserslautern, Germany, October 8–11 1996. Chapman and Hall. FORTE/PSTV'96: Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols, and Protocol Specification, Testing, and Verification.

[65] Eugene J. Rollins and Jeannette M. Wing. Specifications as search keys for software libraries. In *International Conference on Logic Programming*, Paris, June 1991.

[66] James B. Saxe, Stephen J. Garland, John V. Guttag, and James J. Horning. Using transformations and verification in circuit design. In J. Staunstrup and R. Sharp, editors, *International Workshop on Designing Correct Circuits*. North-Holland, IFIP Transactions A-5, 1992. Also published as DEC Systems Research Center Report 78, September 1991. Reprinted in [61]. Superseded by [67].

[67] James B. Saxe, Stephen J. Garland, John V. Guttag, and James J. Horning. Using transformations and verification in circuit design. *Formal Methods in System Design*, 3(3):181–209, December 1993.

[68] Elizabeth A. Scott and K. J. Norrie. Using LP to study the language PL. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[69] Jørgen F. Søgaard-Anderson, Stephen J. Garland, John V. Guttag, Nancy A. Lynch, and Anna Pogosyants. Computed-assisted simulation proofs. In Costas Courcoubetis, editor, *Fifth Conference on Computer-Aided Verification*, pages 305–319, Heraklion, Crete, June 1993. Springer-Verlag Lecture Notes in Computer Science 697.

[70] Jørgen A. Staunstrup, Stephen J. Garland, and John V. Guttag. Localized verification of circuit descriptions. In *International Workshop on Automatic Verification Methods for Finite State Systems*, pages 349–364, Grenoble, June 1989. Springer-Verlag Lecture Notes in Computer Science 407.

[71] Jørgen A. Staunstrup, Stephen J. Garland, and John V. Guttag. Mechanized verification of circuit descriptions using the Larch Prover. In V. Stavridou, T. F. Melham, and R. T. Boute, editors, *Theorem Provers in Circuit Design*, pages 277–299, Nijmegen, The Netherlands, June 1992. IFIP TC10/WG10.2, North-Holland, IFIP Transactions A-10.

[72] Yang Meng Tan. Semantic analysis of Larch interface specifications. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[73] Yang Meng Tan. *Formal Specification Techniques for Promoting Software Modularity, Enhancing Documentation, and Testing Specifications*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, May 1994. MIT/LCS/TR–619.

[74] Yang Meng Tan. Interface language for supporting programming styles. In *POPL Workshop on Interface Definition Languages*, Portland, Oregon, January 1994.

[75] Yang Meng Tan. *Formal Specification Techniques for Engineering Modular C*, volume 1. Kluwer International Series in Software Engineering, Boston, 1995.

[76] David S. Taylor. A beginner's strategy guide to the Larch Prover. S. B. Thesis, Department of Electrical Engineering and Computer Science, MIT, May 1990.

[77] Mark T. Vandevoorde. Optimizing programs with partial specifications. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[78] Mark T. Vandevoorde. Specifications can make programs run faster. In *Proceedings of TAPSOFT '93*. Springer-Verlag, April 1993.

[79] Mark T. Vandevoorde. Exploiting specifications to improve program performance. Technical Report LCS/TR–598, MIT, February 1994. Ph. D. Thesis, Department of Electrical Engineering and Computer Science.

[80] Mark T. Vandevoorde and John V. Guttag. Using specialized procedures and specification-based analysis to reduce the runtime costs of modularity. In *Symposium on the Foundations of Software Engineering*, December 1994.

[81] Mark T. Vandevoorde and Deepak Kapur. Distributed Larch Prover (DLP): an experiment in parallelizing a rewrite-rule based prover. In *Rewriting Techniques and Applications*, July 1996.

[82] Mary A. Vogt. Extension of the Larch Prover by a method of inference using linear arithmetic. S. B. Thesis, Department of Electrical Engineering and Computer Science, MIT, September 1990.

[83] Frederic Voisin. A new front-end for the Larch Prover. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[84] Jeannette M. Wing. A two-tiered approach to specifying programs. Technical Report LCS/TR–299, MIT, May 1983. Ph. D. Thesis, Department of Electrical Engineering and Computer Science.

[85] Jeannette M. Wing. A Larch specification of the library problem. In *Fourth International Workshop on Software Specifications and Design*, pages 34–41, Monterey, April 1987. Also available as CMU–CS–86–168, December 1986.

[86] Jeannette M. Wing. Writing Larch interface language specifications. *ACM Transactions on Programming Languages and Systems*, 9(1):1–24, January 1987.

[87] Jeannette M. Wing. Specifying Avalon objects in Larch. In *International Joint Conference on Theory and Practice of Software Development, TAPSOFT*, pages 61–80, Barcelona, March 13–17 1989. Springer-Verlag Lecture Notes in Computer Science 352. Also available as CMU–CS–88–208, December 1988.

[88] Jeannette M. Wing. Specifying recoverable objects. In *Sixth Annual Northwest Software Quality Conference*, pages 317–331, Portland, OR, September 1990. Also available as CMU–CS–88–170, August 1988.

[89] Jeannette M. Wing. Using Larch to specify Avalon/C++ objects. *IEEE Transactions on Software Engineering*, 16(9):1076–1088, September 1990. Revised version in [87].

[90] Jeannette M. Wing and Chun Gong. Machine-assisted proofs of properties of Avalon programs. TR CMU–CS–89–171, Carnegie Mellon University, August 1989.

[91] Jeannette M. Wing and Chun Gong. Experience with the Larch Prover. In *Workshop on Formal Methods in Software Development*, pages 140–143, May 1990.

[92] Jeannette M. Wing, Eugene Rollins, and Amy Moormann Zaremski. Thoughts on a Larch/ML and a new application for LP. In Ursula Martin and Jeannette M. Wing, editors, *First International Workshop on Larch*. Springer-Verlag, July 1992.

[93] Jeannette M. Wing and Amy Moormann Zaremski. A formal specification of a visual language editor. In *Sixth International Workshop on Software Specification and Design*, Como, Italy, October 1991. Also available as CMU–CS–91–112, February 1991.

[94] Jeannette M. Wing and Amy Moormann Zaremski. Unintrusive ways to integrate formal specifications in practice. In S. Prehn and W. J. Toetenel, editors, *VDM91: Formal Software Development Methods*, Delft, October 1991. Springer-Verlag Lecture Notes in Computer Science 551.

[95] Amy Moormann Zaremski. A Larch specification of the Miro editor. Technical Report CMU–CS–91–112, Carnegie Mellon University, February 1991.