Course Information and Syllabus

This is an introductory course on formal software specification and verification, covering various formalisms, methods, and tools for specifying the properties of a software program and for verifying that the program meets its specification. We will focus on deductive (theorem proving) methods. A separate, complementary course entitled "Automatic Verification" covers algorithmic (model checking) methods.

Instructor

Yih-Kuen Tsay (蔡益坤), Room 1108, Management II, 3366-1189, `tsay@im.ntu.edu.tw`

Lectures

Wednesday 9:10AM–12:10PM, Room 302, College of Management, Building II

Office Hours

Wednesday 1:30–2:30PM (Room 1108, Management II) or by appointment

Prerequisites

Computer Programming and Discrete Mathematics

Textbook

Class Notes and *Selected Readings*

Syllabus/Schedule

We shall seek to strike a balance between depth and breadth, covering both the foundations and some of the more successful formalisms, techniques, and tools. Below is a tentative list of topics and their schedule:

- Introduction (.5 week: 09/17a)
- Propositional and First-Order Logics (1.5 weeks: 09/17b, 09/24)
- Logical Proofs in the Coq Proof Assistant (1 week: 10/01)
- Verification of Sequential Programs: Hoare Logic (2 weeks: 10/08, 10/15)
- Predicate Transformers and Program Derivation (1 week: 10/22)
- Semantic Modeling in Coq (1 week: 10/29)
- Procedures + Object Orientation (1 week: 11/05)
- Program Verification Tools: Why, Caduceus, and Krakatoa (1 week: 11/12)
- Data Refinement + Formal Methods: Z, B, and Alloy (3 weeks: 11/19, 11/26, 12/03)
- Concurrent, Reactive Systems: Owicki-Gries Method, UNITY, Linear Temporal Logic (2 weeks: 12/10, 12/17)
- Selected Topics: Modular/Compositional Reasoning (1 week: 12/24)
- **Final** (**2008/12/31**)

- Selected Topics: Separation Logic (1 week: 2009/01/07)
- Selected Topics: Proof-Carrying Code (1 week: 2009/01/14)

Grading

Homework Assignments 20%, Final 40%, Term Paper/Report 40%

Web Site

`http://www.im.ntu.edu.tw/~tsay/courses/ssv/`

References

[1] *Logic for Computer Science*, J.H. Gallier, Harper & Row Publishers, 1985. (free!)

[2] *Proof Theory and Automated Deduction*, J. Goubault-Larrecq and I. Mackie, Kluwer Academic Publishers, 1997.

[3] *A Logical Approach to Discrete Math*, D. Gries and F.B. Schneider, Springer-Verlag, 1993.

[4] *Foundations for Programming Languages*, J.C. Mitchell, The MIT Press, 1996.

[5] *Formal Syntax and Semantics of Programming Languages*, K. Slonneger and B.L. Kurtz, Addison-Wesley, 1995.

[6] *Verification of Sequential and Concurrent Programs, 2nd Edition*, K.R. Apt and E.-R. Olderog, Springer-Verlag, 1997.

[7] *The Science of Programming*, D. Gries, Springer-Verlag, 1981.

[8] *Predicate Calculus and Program Semantics*, E.W. Dijkstra and C.S. Scholten, Springer-Verlag, 1990.

[9] *Programming from Specifications, 2nd Edition*, C. Morgan, 1994.

[10] *The Z Notation: A Reference Manual, 2nd Edition*, J.M. Spivey, 1992. (free!)

[11] *Software Engineering with B*, J.B. Wordsworth, Addison-Wesley, 1996.

[12] *Software Abstractions: Logic, Language, and Analysis*, D. Jackson, MIT Press, 2006.

[13] *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Z. Manna and A. Pnueli, Springer-Verlag, 1992.

[14] *Temporal Verification of Reactive Systems: Safety*, Z. Manna and A. Pnueli, Springer, 1995.

[15] *Temporal Verification of Reactive Systems: Progress*, Z. Manna and A. Pnueli, Book Draft, 1996. (free!)

[16] *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*, L. Lamport, Addison-Wesley, 2003.

[17] *Parallel Program Design: A Foundation*, K.M. Chandy and J. Misra, Addison-Wesley, 1988.

[18] *A Discipline of Multiprogramming: Programming Theory for Distributed Applications*, J. Misra, Springer, 2001

[19] *Beauty Is Our Business: A Birthday Salute to Edsger W. Dijkstra*, Edited by W.H.J. Feijen, A.J.M. van Gasteren, D. Gries, and J. Misra, Springer-Verlag, 1990

[20] *The Formal Methods Page*: `http://vl.fmnet.info/`, J. Bowen. (Note: this Web portal provides links to numerous formal methods and tools.)