Computing and Software 701

Logic and Discrete Mathematics In Software Engineering

Fall 2005

Presentation Topics

Revised: 28 September 2005

1. Present Imre Lakatos's method of proofs and refutations.
2. Present a sound and complete natural deduction system for propositional logic.
3. Present a sound and complete semantic tableau system for propositional logic.
4. Present a sound and complete Gentzen system for propositional logic.
5. Present a sound and complete resolution system for propositional logic.
6. Explain what binary decision diagrams are and what they are used for.
7. Present fuzzy propositional logic.
8. Present constructive propositional logic.
9. Explain what the ordinals are.
10. Explain what the hyperreal numbers are.
11. Explain what the surreal numbers are.
12. Present one of the set-theoretic paradoxes other than Russell's Paradox.
13. Present a proof of the Schröder-Bernstein Theorem.
14. Explain what a Boolean algebra is and give two examples of Boolean algebras.

15. Describe what "many-sorted" first-order logic is.

16. Present a sound and complete natural deduction system for first-order logic.

17. Present a sound and complete semantic tableau system for first-order logic.

18. Present a sound and complete Gentzen system for first-order logic.

19. Present a sound and complete resolution proof system for first-order logic. Explain why proof systems of this kind are employed in automated theorem provers like Otter.

20. Show that monadic first-order logic is decidable. That is, let $L = (\emptyset, \emptyset, \mathcal{P})$ where $\mathcal{P}$ is a finite set of unary predicate symbols, and then show that the problem of whether or not a formula of L is valid is decidable.

21. Define the notion of an isomorphism between two models of a language of FOL. Explain what it means for two models to be "equal up to isomorphism".

22. Explain what logic programming is and briefly discuss how it is implemented in Prolog.

23. Present Ackermann's function.

24. Present the Knaster-Tarski fixed point theorem for complete partial orders.

25. Present the fixed point theorem for continuous functionals.

26. Present the fixed point method for defining recursive functions described in W. M. Farmer, "A scheme for defining partial higher-order functions by recursion", in: A. Butterfield, ed., *3rd Irish Workshop on Formal Methods (Galway, Ireland, July 1999)*, 13 pp., *electronic Workshops in Computing*, Springer-Verlag, `http://ewic.bcs.org/conferences/ 1999/3rdirish/papers/ paper5.htm`, 1999. This paper is available at

    `http://imps.mcmaster.ca/doc/rec-def-abs.txt`

27. Present Church's lambda calculus.

28. Present Curry's combinatory logic.
29. Explain what Hilbert's ϵ operator is and show how it can be used to define the quantifiers $\forall$ and $\exists$.
30. Give a quick introduction to category theory.
31. Explain what expert systems are and the role logic plays in them.
32. Give an introduction to CTL (Computational Tree Logic).
33. Give an introduction to ITL (Interval Temporal Logic).
34. Show how to compute the complement of an automaton.
35. Show how to minimize an automaton.
36. Give an introduction to petri nets.
37. Show the Kripke model of the Dining Philosophers Problem. Formulate the properties in LTL.
38. Present the elevator specification developed at CMU; express it written in temporal logic.
39. Give an introduction to Statecharts (you can refer to the UML version).
40. Give an introduction to SRI's Symbolic Analysis Laboratory (SAL), a symbolic model checking tool.