Computing and Software 760
Logic for Practical Use
McMaster University, Winter 2010

Exercises

Revised: 27 February 2010

Requirements

Below is a list of exercises (that will grow over the course of the term). You are required to complete 24 of them according to the following schedule:

1. Eight exercises including Exercises 1–5 must be submitted by March 4.
2. Another eight exercises must be submitted by March 25.
3. A final eight exercises must be submitted by April 15.

Exercise List

1. Let $T = (L, \Gamma)$ be a theory in first-order logic, and let $\mathbf{P}$ be a sound and complete proof system for L. Prove that T is consistent in $\mathbf{P}$ iff T is satisfiable.

2. The slides for 02 Three Traditional Logics slides include a presentation of the syntax and semantics of FOL and STT. Using these presentations as models, give the syntax and semantics of *many-sorted first-order logic (MS-FOL)*.

3. Formalize in FOL a theory of an abstract bipartite graph.

4. Formalize in MS-FOL a theory of an abstract bipartite graph.

5. Let **Nat** be a theory in MS-FOL that includes the usual machinery of natural number arithmetic (a sort nat, individual constants $0, 1, 2, \ldots$, binary function symbols $+$ and $*$, and binary predicate symbols $<$ and $\leq$. Define an extension **Address** of **Nat** that contains a sort $\mathsf{address}$ of values composed of four octets (i.e., 8-bit natural numbers).

6. Do Exercise 5 in MS-STT.

7. Do Exercise 5 in ZF.

8. Define an extension **Monoid** of **Nat** that formalizes the notion of an abstract monoid. **Monoid** should include a sort monoid, an individual constant zero, and a binary function symbol add (written infix). Define in **Monoid** the concept of iterating the add operation specified by:

$$\Sigma(m, n, f) = \begin{cases} \Sigma(m, n-1, f) \ \mathsf{add} \ f(n) & \text{if } m \leq n \\ \mathsf{zero} & \text{if } m > n \end{cases}$$

$\Sigma(m, n, f)$ would usually be written in informal mathematics as

$$\sum_{i=m}^{n} f(i).$$

9. Do Exercise 8 in MS-STT.

10. Do Exercise 8 in ZF.

11. Define in MS-FOL a theory **Monoid-2** of two abstract monoids. Define in **Monoid-2** the notion of a homomorphism from the first monoid to the second and the notion of the kernel of such a homomorphism.

12. Do Exercise 11 in MS-STT.

13. Do Exercise 11 in ZF.

14. Let T be any theory of MS-FOL. Define in T an inductive data of the natural numbers. The sort should be nat, the constructors zero and suc, and the single selector pred.

15. Do Exercise 14 in MS-STT.

16. Do Exercise 14 in ZF.

17. Let T be any extension of **Nat** in MS-FOL. Assume T contains a sort α. Define in T an inductive data of the stacks of elements of α. The sort should be stack, the constructors bottom and push, and the selectors top and pop.

18. Do Exercise 17 in MS-STT.

19. Do Exercise 17 in ZF.

20. Formalize in FOL a theory of well orders.

21. Do Exercise 20 in MS-STT.

22. Do Exercise 20 in ZF.

23. Prove that every Goodstein sequence s eventually reaches 0 (i.e., that there is some $n \geq 0$ such that, for all $n' \geq n$, $s(n') = 0$).

24. Using the Compactness Theorem, prove that every partial order on a set S can be be extended to a total order on S

25. Using the Compactness Theorem, prove that first-order Peano arithmetic has a nonstandard model (i.e., a model with infinite elements).

26. Using the Compactness Theorem, prove that any first-order theory of the real numbers has a model that contains a positive infinitesimal.

27. Using the Compactness Theorem, prove that any first-order theory that has an infinite model has infinitely many nonisomorphic infinite models. (This is a version of the Upward Löwenheim-Skolem Theorem.)

28. Using the presentations of FOL and STT in the slides as a model, give the syntax and semantics of *first-order logic with definite description*.

29. Using the presentations of FOL and STT in the slides as a model, give the syntax and semantics of *first-order logic with indefinite description*.

30. In STT define function abstraction $(\lambda\, x : \alpha \,.\, B)$ as a notational definition using definite description and function application.

31. In the STT theory **PA** define addition explicitly without using recursion. (Hint: Use definite description.)

32. In the STT theory **COF** define a predicate representing the natural numbers explicitly without using recursion. (Hint: Use definite description.)

33. Let STTwID be STT plus indefinite description $(\epsilon\, x : \alpha \,.\, B)$. Given a type $\alpha \in \mathcal{T}$, define in STTwID a choice operator choice_α of type $((\alpha \rightarrow *) \rightarrow \alpha)$ for sets of elements in D_α.

34. Show how the Empty Set axiom of ZF can be proved from the other axioms of ZF.

35. Replacement is formalized in ZF using a sentence schema, but it can be formalized in NBG with a single sentence because every function on sets can be represented by a class. Write a sentence in NBG that formalizes replacement.

36. What changes would have to be made to STT so it handles the undefinedness of improper definite descriptions using the Unspecified Values approach to undefinedness instead of the Internal Error Values approach?

37. What changes would have to be made to FOLwU to allow individual constants to be undefined?

38. Which of the six approaches to handling partial functions and undefinedness in traditional logics do you prefer? Explain your reasoning.