COM3412: Logic and Computation

Sample solutions and marking guidelines

Summer 2008

1. (a) *First-order language:* The set of formulae that can be constructed using the standard logical constants of first-order logic (i.e., $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$, $\forall$ and $\exists$) and a non-logical vocabulary consisting of specified sets of individual constants (possibly empty), function symbols (possibly empty) and predicate symbols (non-empty).

(3 marks)

Interpretation: A domain Δ is specified, and the non-logical vocabulary of the language is interpreted over the domain according to the following scheme:

- Individual constants are interpreted as referring to elements of Δ;
- Function symbols are interpreted as referring to functions on Δ, a function symbol of arity n being associated with a function of type $\Delta^n \rightarrow \Delta$;
- Predicate symbols are interpreted as relations on Δ, a predicate symbol of arity n being associated with a subset of Δ^n.

(4 marks)

Model: A model for a set of formulae is an interpretation under which all the formulae in the set are assigned the value 'true' by the recursive evaluation procedure determined by that interpretation.

(3 marks)

Validity: An inference from premiss-set Σ to conclusion ϕ is valid if and only if every model for Σ satisfies ϕ.

(2 marks)

(b) Introduction rules are used to infer a formula containing the given logical constant from formulae and/or subderivations which do not contain that particular occurrence of the logical constant.

(2 marks)

Elimination rules are used to infer from a formula containing the given logical constant a formula which does not contain that occurrence of the logical constant.

(2 marks)

Logical constants are those symbols (notably connectives and quantifiers) whose interpretation is fixed by the logic and hence is invariant across all interpretations of the language under consideration.

(2 marks)

(c) Semantic rule for $\vee$: $\phi \vee \psi$ is true iff and only if at least one of ϕ and ψ is true.

(2 marks)

Semantic rule for $\rightarrow$: $\phi \rightarrow \psi$ is false iff ϕ is true and ψ is false.

(2 marks)

Justification of $\rightarrow$-introduction: Suppose we have $\Sigma \cup \{\phi\} \models \psi$, and let M be any model for Σ. If M satisfies ϕ, then M is a model for $\Sigma \cup \{\phi\}$ and therefore satisfies ψ; from the semantic rule for $\rightarrow$, this means that M satisfies $\phi \rightarrow \psi$. If, on the other hand, M satisfies $\neg\phi$, then by the semantic rule for $\rightarrow$, this again means that M satisfies $\phi \rightarrow \psi$. Since one of the two cases must hold, M satisfies $\phi \rightarrow \psi$. Hence $\Sigma \models \phi \rightarrow \psi$, justifying $\rightarrow$-introduction.

(3 marks)

Justification of $\vee$-introduction: Suppose we have $\Sigma \cup \{\phi\} \models \chi$ and $\Sigma \cup \{\psi\} \models \chi$, and let M be any model for $\Sigma \cup \{\phi \vee \psi\}$. By the semantic rule for $\vee$, M is either a model for $\Sigma \cup \{\phi\}$ or a model for $\Sigma \cup \{\psi\}$. In either case, we infer that M is a model for $\Sigma \cup \{\chi\}$. Hence $\Sigma \cup \{\phi \vee \psi\} \models \chi$, justifying $\vee$-elimination.

(3 marks)

(d)

1. $A \rightarrow C$	(Premiss)
2. $B \rightarrow C$	(Premiss)
3. SUBDERIVATION	
3.1. $A \vee B$	(Assumption)
3.2. SUBDERIVATION	
3.2.1. A	(Assumption)
3.2.2. C	(1, 3.2.1, $\rightarrow$-elim)
3.3. SUBDERIVATION	
3.3.1. B	(Assumption)
3.3.2. C	(2, 3.3.1, $\rightarrow$-elim)
3.4. C	(3.1, 3.2, 3.3, $\vee$-elim)
4. $A \vee B \rightarrow C$	(3, $\rightarrow$-intro)

(12 marks)

2. (a) A proof system for a logic is a finitely-specifiable set of rules for constructing proofs, i.e., demonstrations completed in a finite number of steps that some inference is either valid or invalid in the logic.

(2 marks)

Examples include: for Propositional Calculus, truth tables; for either Propositional or Predicate Calculus, truth trees (semantic tableaux), Natural Deduction, resolution.

(4 marks)

(i.e., for each of two examples: 1 mark for naming it, 1 mark for brief description)

Soundness: Every inference the proof system certifies as valid is in fact valid.

(2 marks)

Completeness: Every inference that is in fact valid is certified by the proof system as valid.

(2 marks)

Decidability: There is an algorithm (effective procedure) which can reliably determine, for any inference, whether or not it is valid.

(2 marks)

(b) Suppose we have a sound and complete proof system for the logic, and suppose that ϕ is entailed by some set of formulae Σ. Since the proof system is complete, there is a proof of $\Sigma \models \phi$ in the logic. Since the proof consists of finitely many steps, at most finitely many formulae in Σ appear in the proof. Let $\Sigma_0 \subseteq \Sigma$ be the (finite) set of all formulae in Σ which appear in the proof. Then the proof is also a proof of $\Sigma_0 \models \phi$. Since the proof system is sound, ϕ is entailed by Σ_0. We have shown that whenever a formula ϕ is entailed by a set of formulae Σ, it is entailed by some finite subset of Σ. Hence the logic is compact.

(8 marks)

(c) The inference

$$\{\phi_i \mid i \in \mathbb{N}\} \models \bigwedge_{i=0}^{\infty} \phi_i$$

is valid, since the conclusion is true iff all the ϕ_i are true; and these are precisely the premises of the inference.

If the logic were compact, then there would be some finite subset Σ_0 of $\{\phi_i\}$ which entails the conclusion. But this cannot be correct, since we can find a model of Σ_0 which falsifies at least one of the $\phi_i \notin \Sigma_0$. Hence the logic is not compact.

[Note: The argument as stated is not fully rigorous since it assumes the ϕ_i are sufficiently logically independent. The marker should not be too fussy about this point — a student who can produce something like the above has already shown sufficient understanding.]

(10 marks)

3. (a) The Halting Problem: Given an arbitrary Turing machine M and input tape t, to determine whether or not M will halt when run with t.

(3 marks)

The Decision Problem for First-order Logic: Given an arbitrary set of formulae of first-order logic, and an arbitrary formula, to determine whether the latter is a logical consequence of the former. [Alternative version: Given an arbitrary set of first-order formulae, to determine whether or not it is satisfiable, i.e., whether there is an interpretation under which all the formulae in the set come out true.]

(3 marks)

The practical importance of the Halting Problem is that since computer programs are equivalent to Turing machines, the Halting Problem applies to them too. Non-termination is an obviously undesirable feature for a program intended to produce a specific result, and it would be useful to be able to detect it in advance. Non-termination manifests itself in practice when the system "freezes" and will not respond to further input.

(3 marks)

The practical importance of the Decision Problem is that many everyday information processing problems can be formulated as first-order inferences, so if we had a general method of testing such inferences for validity then we would be able to solve such problems, and maybe get computers to do more of our reasoning for us.

(3 marks)

(b) A Turing machine together with its tape, as well as the step-by-step operation of the machine, can be comprehensively described by first-order formulae. The machine halts if and only if the formula stating that it halts after some number of steps is a logical consequence of the formulae describing the machine and it operation. Hence if we can solve the Decision Problem, we can solve the Halting Problem as well.

(6 marks)

Conversely, given a first-order inference we could use a method like Truth Trees to test it for validity. If the inference is valid, then this process will always terminate, but if it is invalid it may or may not terminate. If we can solve the Halting Problem then we can test this process for termination first; if it is non-terminating, then we know the inference is invalid, whereas if it terminates we can run it to determine whether or not the inference is invalid. Hence if we can solve the Halting Problem, we can solve the Decision Problem as well.

(6 marks)

(c) We know that we cannot use a Turing Machine to solve the Halting Problem (by the standard *reductio ad absurdum* argument), and so if the Church-Turing thesis is correct, there is no effective procedure for solving it. By the equivalence of the two problems, it follows that there is no effective procedure for solving the Decision Problem either. Only if we were to discover some method of computation which was (a) effective and (b) non-Turing Equivalent, would there be any chance of finding a general method for solving either of these problems.

(6 marks)

4. (a) In Ex1, i must be 0, since for real numbers, $x + y = x$ if and only if $y = 0$.
In Ex2, i must be 1, since for real numbers, $x.y = x$ if and only if $y = 1$.
In Ex3, i must be $\emptyset$, since $\emptyset$ is the only set y for which we *always* have $x \cup y = x$.

(4 marks)

(b) In (Ex1), (4) is not satisfied since, e.g., $2 + 2 \neq 2$.
In (Ex2), (4) is not satisfied since, e.g., $2 \times 2 \neq 2$.
In (Ex3), (4) is satisfied, since for any set X we have $X \cup X = X$.

(4 marks)

(c)
$$\begin{array}{rcll} f(x, f(x, y)) & = & f(f(x, x), y) & \text{(by 2)} \\ & = & f(x, y) & \text{(by 4)} \\ & = & f(y, x) & \text{(by 1)} \\ & = & f(f(y, y), x) & \text{(by 4)} \\ & = & f(y, f(y, x)) & \text{(by 2)} \end{array}$$

(6 marks)

(d) In (Ex1), (5) is satisfied since for any x we have $x + (-x) = 0$.
In (Ex2), (5) is not satisfied, since there is no y such that $0 \times y = 1$.
In (Ex3), (5) is not satisfied, since if $X \neq \emptyset$ then there is no set Y such that $X \cup Y = \emptyset$.

(4 marks)

(e)
$$\begin{array}{rcll} f(x, f(x, y)) & = & f(f(x, x), y) & \text{(by 2)} \\ & = & f(x, y) & \text{(by 4)} \\ & = & i & \text{(by assumption)} \\ f(x, f(x, y)) & = & f(x, i) & \text{(by assumption)} \\ & = & x & \text{(by 3)} \end{array}$$
By (5), for each x there is a y such that $f(x, y) = i$, so by the above argument, $x = f(x, f(x, y)) = i$, so all elements are equal to i, i.e., the domain contains only one element.

(6 marks)

(f) A complete axiomatisation of a first-order theory is a set of formulae whose logical consequences are precisely the formulae of the theory.
If $\{1, 2, 3\}$ were a complete axiomatisation of some theory, it would imply whichever of 4 and $\neg 4$ is in the theory. But (Ex1) satisfies $\{1, 2, 3, \neg 4\}$, while (Ex3) satisfies $\{1, 2, 3, 4\}$, so $\{1, 2, 3\}$ implies neither 4 nor $\neg 4$. [We could use 5 here instead of 4.]
The set $\{1, 2, 3, 4, 5\}$ only holds for a domain $\{i\}$ in which $f(i, i) = i$. All such domains are isomorphic, so we have a complete axiomatisation.

(6 marks)

5. (a) For first-class marks I would expect to see:
   - a clear statement of what exactly Cook proved (this would also involve giving a correct definition of the class NP);
   - an accurate outline of the method by which he proved it;
   - a correct definition of the class NP-complete;
   - a discussion of NP-complete problems and their implications for practical computing.

   (b) For first-class marks I would expect to see:
   - a clear statement of what the Church-Turing Thesis states, with an appreciation of the fact that it is not a mathematical theorem, and as such not susceptible to rigorous proof;
   - an account of the reasons why the thesis is generally held to be true, and the implications this has for practical computing;
   - a discussion of what would have to happen for the thesis to come to be discredited.

(30 marks)