Final Exam

(100 points)

Honor Code: Each question is worth 10 points. There is one bonus question worth 5 points. In contrast to the homework assignments, you may **not** collaborate on this final exam. You may not discuss the exam with anybody but the TAs and the instructor, who will only answer clarification questions. You may use no books other than the Enderton textbook and the notes on modal logic. Good Luck!

1. (10 points) For each of the following parts, give an example of a theory T with the given property or explain why there is none:

   (a) T is complete

   (b) T is not complete

   (c) T has only finite models

   (d) T has only infinite models

   (e) T has only countably infinite models

2. Consider the first-order language with equality and one binary predicate symbol '$<$'. Let Δ be the set of sentences expressing that $<$ is a dense linear order without endpoints. That is, Δ consists of the five sentences given on page 159 in Enderton.

   (a) (10 points) Using Cantor's Theorem, prove that if $\mathcal{A}$ and $\mathcal{B}$ are dense linear orderings without endpoints, then $\mathcal{A}$ and $\mathcal{B}$ are elementarily equivalent.

3. (10 points) Recall the liar's paradox "This sentence is false" which (suitably formalized) has been put to much use in the logic and set theory. The following result, first proved by Alfred Tarski, has been used to argue that we cannot give a coherent semantics for any language that, like English, contains its own truth predicate.

   Suppose that $\mathcal{L}$ is a first-order language with at least one constant symbol and that $\mathcal{L}^+$ is the result of adding a new unary predicate symbol T to $\mathcal{L}$. We say that a structure $\mathcal{B}$ for $\mathcal{L}^+$ **extends** a structure $\mathcal{A}$ for the language $\mathcal{L}$ provided $|\mathcal{B}| = |\mathcal{A}|$ and for each constant symbol, predicate symbol and function symbols S of $\mathcal{L}$, we have $S^{\mathcal{B}} = S^{\mathcal{A}}$. [So, when $\mathcal{B}$ extends $\mathcal{A}$, $\mathcal{B}$ interprets the language $\mathcal{L}$ in the same way that $\mathcal{A}$ does and $\mathcal{B}$ also interprets the new symbol T]. We say that a structure $\mathcal{A}$ for $\mathcal{L}$ is a **ground structure** provided every every sentence of $\mathcal{L}^+$ is a member of $|\mathcal{A}|$. Finally, given a ground structure $\mathcal{A}$ for $\mathcal{L}$ and an extension $\mathcal{B}$ of $\mathcal{A}$, we say that T is a **truth predicate** for $\mathcal{B}$ provided, for every sentence φ of $\mathcal{L}^+$, we have $\mathcal{B} \models Tx[\![\varphi]\!]$ iff $\mathcal{B} \models \varphi$, i.e., Tx is true in $\mathcal{B}$ with φ assigned to x iff φ is true in $\mathcal{B}$.

Show that there is a ground model $\mathcal{A}$ that cannot be extended to a model $\mathcal{B}$ for which T is a truth predicate.

4. Definitions appear in almost any field of inquiry. Some of the techniques we learned this quarter allow us to analyze more precisely what constitutes a definition and what different kinds of definitions there are. In this exercise we look at three types of definitions that came up during the quarter and their interconnections: 1. defining a relation or function in a fixed structure, 2. an explicit definition in the language, and 3. a recursive definition.[1]

   To simplify things, we will only consider the case of defining a new function symbol in terms of other function symbols, but the story can be extended to relation symbols as well.

   First, recall the notion of *definability in a structure* that we discussed in class (recall Enderton pg. 90). We can simplify that definition for the case of functions and terms as follows: consider a structure $\mathcal{A}$. Any term $t(v_1, \ldots, v_n)$ with free variables among $\{v_1, \ldots, v_n\}$ **defines an n-ary functions** $F_t : |\mathcal{A}|^n \to |\mathcal{A}|$ in $\mathcal{A}$ as follows: $F_t(a_1, \ldots, a_n) = t^{\mathcal{A}}[a_1, \ldots, a_n]$. Another way to state this is

$$F_t(a_1, \ldots, a_n) = \overline{s(v_1|a_1, v_2|a_2, \ldots, v_n|a_n)}(t)$$

   where s is a substitution. In other words, terms t with n-free variables define n-ary functions on $\mathcal{A}$.

   While this notion of definability is useful, it is limited to specific structures and doesn't quite capture that we normally use definitions when introducing new symbols to the language. The following notion overcomes this limitation. It captures we we do when we define a new function in mathematics with respect to some background knowledge by saying "let $f(x, y) = \alpha(x, y)$" where $\alpha(x, y)$ is some expression not involving f.

   Let T be a theory in some language $\mathcal{L}$ and let f be some n-ary function symbol in $\mathcal{L}$. Let $\mathcal{L}_0$ be $\mathcal{L} - \{f\}$ (so, $\mathcal{L}$ extends $\mathcal{L}_0$ by adding the functions symbol f). Then we say that a term $t(v_1, \ldots, v_n)$ in $\mathcal{L}_0$ **defines f explicitly in T** iff

$$\forall v_1 \forall v_2 \cdots \forall v_n (f(v_1 v_2 \cdots v_n) = t(v_1, \ldots, v_n)) \in T$$

   The crucial thing is that $t(v_1, \ldots, v_n)$ does not involve f.

   (a) (10 points) Consider the languages $\mathcal{L}$ and $\mathcal{L}_0$ as above with f in $\mathcal{L}$ and t is $\mathcal{L}_0$. Let $\mathcal{A}$ be a structure in the language $\mathcal{L}$. Show that t defines a functions $f^{\mathcal{A}}$ in $\mathcal{A}$ iff t explicitly defines f in $\mathsf{Th}(\mathcal{A})$.

   It seems that these two notions of definability really amount to the same thing after all. So, what is the use of the notion of "explicit definability"? An example will illustrate

[1] In particular, the following questions focus on the deference between explicitly definitions like "x is defined as the smallest prime number bigger than 1000" and recursive definitions like "$\neg p$ is true iff p is not true".

its usefulness. Consider the first-order language with equality, a constant symbol $\mathbf{0}$, a unary functions symbol $\mathbf{S}$ and a binary function symbol $+$. Let $\mathcal{N}$ be the standard model of arithmetic for this language, where the domain is $\mathbb{N}$, $\mathbf{S}^{\mathcal{N}}$ is the successor function, $\mathbf{0}^{\mathcal{N}}$ is 0 and $+^{\mathcal{N}}$ is addition. Let T consist of the following four sentences:

1. $\forall x(\mathbf{S}x \neq \mathbf{0})$
2. $\forall x \forall y(\mathbf{S}x = \mathbf{S}y \rightarrow x = y)$
3. $\forall x(x + \mathbf{0} = x)$
4. $\forall x \forall y((x + \mathbf{S}y) = \mathbf{S}(x + y))$

Note that T contains a recursive definition of addition in terms of successor, and that $\mathcal{N}$ is a model of these axioms. Does the recursive definition also yield an explicit definition?

(b) (10 points) It can be shown that there is no term without the $+$ symbol that defines the addition function in $\mathcal{N}$. Use this fact to show that there is no term (without $+$) that defines $+$ explicitly in $\mathsf{Cn}(T)$.

This raises the question in what way a recursive definition is actually a definition. What the two recursion equations for addition above allow us to do is *compute*. More precisely, call a term **closed** if it contains no variables. Then for any closed term that involves the addition symbol $+$, we can eliminate $+$ and obtain a term that only contains $\mathbf{0}$ and $\mathbf{S}$, which essentially corresponds to a natural number (eg., $\mathbf{SS0}$ corresponds to 2). What is essential here is to consider closed terms.

(c) (10 points) Show that for every closed term t_1 there is a closed term t_2 not containing $+$ such that $t_1 = t_2 \in \mathsf{Cn}(T)$.

Hence, while an explicit definition would allow us to eliminate $+$ in an aribtrary context like $\mathbf{S}x + \mathbf{SS}y$, there recursive definition allows us to eliminate $+$ only for closed terms (eg., $\mathbf{S0}+\mathbf{SS0}$ can be relaced with $\mathbf{SSS0}$ through repeated applications of the recursion equations.)

5. We have already noticed an analogy between the modal operator '$\Box$' and the universal quantifier '$\forall$'. In this question, we will make this connection more formal. There is a precise sense in which the modal language is a *fragment* of the first order language. Let $\mathcal{L}$ be the basic modal language built with $\mathsf{At} = \{p, q, r, \ldots\}$ the set of atomic propositional letters. The first order language $\mathcal{L}_1$ corresponding to $\mathcal{L}$ contains a binary relation symbol $\mathbf{R}$ and for each atomic propositional letter $p \in \mathsf{At}$, a unary predicate symbol $\mathbf{P}$. Any modal model $\mathcal{M} = \langle W, R, V\rangle$ can be viewed as a (first-order) structure for $\mathcal{L}_1$ where the domain of $\mathcal{M}$ is W, the interpretation of $\mathbf{R}$ is the relation R (i.e., $\mathbf{R}^{\mathcal{M}} = R$) and for each unary predicate symbol $\mathbf{P}$ (corresponding to atomic proposition $p \in \mathsf{At}$), $\mathbf{P}^{\mathcal{M}} = \{w \mid V(w, p) = \mathbf{T}\}$.

Now we can (faithfully) translate the modal language into the first-order language. Formally, we define a map $ST : \mathcal{L} \to \mathcal{L}_1$ sending modal formulas to first-order ($\mathcal{L}_1$) formulas with one free variable as follows:

$$\begin{aligned} ST_x(p) &= \mathbf{P}x \\ ST_x(\neg\varphi) &= \neg ST_x(\varphi) \\ ST_x(\varphi \wedge \psi) &= ST_x(\varphi) \wedge ST_x(\psi) \\ ST_x(\Box\varphi) &= \forall y(\mathbf{R}xy \to ST_y(\varphi)) \text{ (where } y \text{ is a new variable)} \\ ST_x(\Diamond\varphi) &= \exists y(\mathbf{R}xy \wedge ST_y(\varphi)) \text{ (where } y \text{ is a new variable)} \end{aligned}$$

For example,

$$ST_x(\Box\Diamond p \wedge \Box\neg q) = \forall x_1(Rxx_1 \to \exists x_2(Rx_1x_2 \wedge Px_2)) \wedge \forall x_3(Rxx_3 \to \neg Qx_3)$$

We first check that this translation preserves truth:

(a) (10 points) Show that for all modal formulas $\varphi \in \mathcal{L}$ and modal models $\mathcal{M} = \langle W, R, V\rangle$, for all states $w \in W$,

$$\mathcal{M}, w \models \varphi \text{ iff } \mathcal{M} \models ST_x(\varphi)[\![x|w]\!]$$

Now, it is clear that not all formulas of $\mathcal{L}_1$ are translation of modal formulas (i.e., of the form $ST_x(\varphi)$ for some modal formula φ). For example, $\forall y(\mathbf{P}x \to \mathbf{R}xy)$ is not of the right syntactic form. However, for some $\mathcal{L}_1$ formulas φ, even if φ is not of the form $ST_x(\psi)$ for some modal formula ψ, φ may be logically equivalent to such a formula.

(b) (10 points) Is $\exists y(\mathbf{R}xy \wedge \neg\mathbf{R}yx \wedge \mathbf{P}y)$ logically equivalent to the standard translation of some modal formula? (i.e., does there exists a modal formuls ψ such that $\exists y(\mathbf{R}xy \wedge \neg\mathbf{R}yx \wedge \mathbf{P}y)$ is logically equivalent to $ST_x(\psi)$?) If yes, provide the modal formula. If the answer is no, explain why.

A natural question is *which formulas of $\mathcal{L}_1$ are equivalent to translations of modal formulas*? The answer to this is beyond the scope of the course (the theorem that answers this question is called the Van Benthem Characterization Theorem and will be discussed in PHIL 154). What is interesting for us is that the standard translation can be used to transfer results about first-order logic to modal logic:

(c) (10 points) Use the standard translation and part (a) above to prove a Löwenheim-Skolem Theorem for modal logic (if a modal formula is satisfiable, then it is satisfiable in a countable model) and a Compactness Theorem.

6. For most of this quarter we worked with first-order languages. The previous question demonstrates that modal logic can be viewed as a fragment of first-order logic. So, the language of modal logic is really "short-hand" for first-order formulas of a certain syntactic shape. In fact, there are many restrictions to the first-order language that

one can imagine. For example, one may be interested in quantifier-free sentences (i.e., propositional sentences), the universal fragment (formulas only containing universal quantifiers), or formulas of the form $\forall x \exists y \varphi(x, y)$ were $\varphi(x, y)$ is quantifier-free. Another restriction that has turned out to be interesting focuses on the number of distinct variables that you allow in a formula. For example, the formula $\exists y \forall x(Rxy \wedge \exists z Ryz)$ uses[2] three distinct variables (x, y and z). After some thought, it is not too hard to realize that there is no need to introduce a new variable z. That is, the formula $\exists y \forall x(Rxy \wedge \exists x Ryx)$ says the same thing with only two variables. In fact, one can show that the standard translation defined in the previous question can be defined making use of only two variables. This shows that modal logic falls inside the two variable fragment of first-order logic[3]

(a) (10 points) Consider the first order language with one binary predicate symbol E and two constant symbols a and b. Models for E can be thought of as (directed) graphs. For each n, we can write a first-order sentence $Path_n(a, b)$ expressing that there is a path of length n from a to b (this is essentially what was needed to solve Enderton #8, pg. 146 from Homework 6). Show that for each n, there is a formula $Path_n(a, b)$ *using only two variables* that states there is a path of length n from a to b. [Note that you will have a different formula for each n, but each formula will use only two variables.]

(b) **Bonus Question** Suppose that $Path^*_n(a, b)$ means "there is a path of length n between a and b and there is no other path between a and b". Show that $Path^*_n(a, b)$ can be defined in first-order logic using only *three* variables.

A fascinating result of Dana Scott shows that the two variable fragment of first-order logic is *decidable* (by contrast, already the three variable fragment of first-order logic is known to be undecidable). Philosophy 152 will study the formal machinery needed to understand this result.

The above questions focused on *fragments* of first-order logic. Much work has also focused on possible *extensions* of first-order logic (for example, see Enderton Chapter 4). The first question that one always asks is whether a particular extension to the first-order language really adds something new. For example, consider the quantifier $\exists! x \varphi(x)$ which means "there is a unique x satisfying φ". It is not hard to see that this is logically equivalent to a formula of first-order logic. So, adding $\exists! x$ to a first-order language does not really add anything new. However, this is not always true. For example, consider $\exists_f x \varphi(x)$ which means that there are *finitely* many elements satisfying φ. Formally, $\mathcal{A} \models \exists_f x \varphi[s]$ iff $|\{a \mid \mathcal{A} \models \varphi[s(x|a)]\}| = n$ where $0 \neq n \in \mathbb{N}$ is a non-zero natural number. In fact, one can show that there is no first order formula

[2]Of course, there are more *occurrences* of variables (in this case 7), but we are interested in the variables used not the occurrences.

[3]The question arises as to whether modal logic is *exactly* the two-variable fragment. That is, is there a first-order formula using only two variables that is not equivalent to the standard translation of some modal formula? The answer to this is yes: $\neg xRx$ is a formula expressing that the current state not related to itself (i.e., it is irreflexive). An argument using bisimulations (as discussed in class) shows that there is not modal formula that expresses that the current state does not have a reflexive arrow).

equivalent to $\exists_f x\varphi(x)$. This also shows that, for example, there is no first-order formula that expresses "there is a finite path from a to b". The raises an interesting question: what makes a logic *first-order*? A fascinating result by Per Lindström shows that first-order logic is the "strongest" logic having the Compactness and Löwenheim-Skolem Theorems.

The midterm is DUE Wednesday, March 18 at noon.

Hints

General Hint: None of the questions require you to do long and complicated constructions. The main challeng is understanding what the question asks and understanding the concepts involved. So if you find yourself going on for pages constructing a particularly ingenious mode, set of sentencees, etc., you are probably on the wrong track, or at least, on an elaborate detour.

1. none
2. You need to use the downward Löwenheim-Skolem Theorem
3. Note that ground models are similar to the term models we looked at when proving the completeness theorm for FOL and ML. In the case of the canonical model and ground models, elements of the domain are expressions from the language. Construct a counter model and use the constant to refer to a liar sentence. Show that T is not a truth predicate.

4 (a) This is really just unpacking the definitions.

4 (b) Assuming that there is no term without the + symbol that defines the addition function in $\mathcal{N}$, prove that there is no term (without +) that defines + explicitly in $\mathsf{Cn}(T)$. Use the previous exercise and relate $\mathsf{Th}(\mathcal{N})$ to $\mathsf{Cn}(T)$.

4(c) Give a proof by induction on t_1. Note that the set of closed terms in this language is inductively generated from the constant $\mathbf{0}$ by the term building operations of + and $\mathbf{S}$. Finally, observe that you are NOT supposed to show that $t_1 = t_2$ is ture in the standard model, but rather that it can be deduced from T. Not all axioms of T may be needed here.

5 (a) Use a proof by induction on the structure of φ.

5 (b) Use the modal invariance lemma (that bisimulations preserve truth).

5 (c) no hint

6 (a) no hint