Final Exam

Problems

The Limits of Logic

Due 4:30pm Friday Dec 8, 2023

Please turn this in either in my mailbox in the Mudd Hall of Philosophy main office, or by email to jeff.russell@usc.edu. (Scans or photos of handwritten pages are fine, as long as they're legible.)

Turn in answers to **five** of the following seven problems, as well as an answer to **one** of the three essay questions.

You may take for granted any of the facts which we proved in class or in the notes. Please state explicitly what facts you are relying on (preferably by giving a brief statement of the fact, or its name if it has one, rather than its number in the textbook).

Problem 1

Consider a signature L containing a one-place function symbol `f`, a name `c`, and no other symbols. Let S be the following L-structure:

- The domain of S is the set of all strings $\mathbb{S}$.
- The name `c` denotes the string `•`. That is,

$$[\texttt{c}]_S = \texttt{•}$$

- The extension of the function symbol `f` is the function that takes each string s to the same string repeated three times.

$$[\texttt{f}]_S(s) = s \oplus \texttt{•••} \quad \text{for each string } s \in \mathbb{S}$$

Prove by induction that, for every L-term t, the denotation of t in S is the same length as t itself.

$$\text{length}\, t = \text{length}\, [\![t]\!]_S$$

Problem 2

Prove the following from the definitions.

(a) For any sets of sentences X and Y and any sentences A and B, if $X \vDash A$ and $Y \vDash B$, then $X, Y \vDash A$ `&` B.

(b) For any set of sentences X and any formula $F(x)$, if $X \cup \{\forall x \; F(x)\}$ is inconsistent, then $X \vDash \exists x \; \neg F(x)$.

Problem 3

For each of the following sets, say whether it is **decidable**, **semi-decidable** (but not decidable), or **undecidable** (and not even semi-decidable). Explain your answers.

(a) The set of all programs that halt.
(b) The complete true theory of strings, Th $\mathbb{S}$.
(c) The set of all truths of first-order logic that are shorter than one million symbols.
(d) The theory that is axiomatized by all of the axioms of the minimal theory of strings S plus a Gödel sentence for S.

Problem 4

Let's call a program A *efficient* if it halts within some number of steps which is less than a billion times the length of A. For a program $A(x)$ that takes an input, say $A(x)$ is efficient for input s iff the program $A\langle s\rangle$ is efficient. We say $A(x)$ is efficient iff it is efficient for every input.

We would like to write a program that can check, for any program, whether it is efficient. In this problem and the next, we will show that no such program can be efficient for all inputs.

Here are some facts you can take for granted for both problems 4 and 5:

(i) The program Diag(x) (from section 7.5 of the book) is efficient. (Remember that this program just sticks together a program with its quotation, which can be done efficiently.)

(ii) The program Flip(x) (also from 7.5) is efficient for the input `False`, and inefficient for the input `True`.

(iii) If $B(x)$ is efficient for the input $[\![A]\!]$, then $B(A)$ (the program that runs A and then B) is also efficient. (Intuitively: if running A and then B is efficient, then if you just skip over running A and provide its result to B directly, this is still efficient.)

(iv) If A is efficient and B is efficient for the input $[\![A]\!]$, then $B(A)$ is efficient. (Intuitively: if you get the output of A efficiently, and then you run $B(x)$ on that output efficiently, then the process of doing both of those things is still efficient overall.)

In this problem, we will prove a new version of Kleene's Fixed Point Theorem about efficiency:

> For any program $F(x)$, there is some program G such that G is efficient iff $F(x)$ is efficient for the input G.

(*Hint.* In fact, you can use the very same program that we used to prove the original version of Kleene's Fixed Point Theorem, $H\langle H(x)\rangle$ where $H(x) = F(\mathrm{Diag}(x))$.)

Problem 5

(This continues from problem 4.)

Suppose there is some program $E(x)$ such that, for any program A,

$$[\![E]\!](A) = \begin{cases} \texttt{True} & \text{if } A \text{ is efficient} \\ \texttt{False} & \text{otherwise} \end{cases}$$

Use what was shown in problem 4 to prove that $E(x)$ is not efficient—that is, there is some program G such that $E\langle G\rangle$ is not efficient.

Hint. Suppose that $E(x)$ is efficient, and use the same reasoning as in the proof of Turing's Theorem to derive a contradiction. You can basically just replace "halts" with "efficient".

Problem 6

Let $L = L_{\mathbb{N}}(\texttt{c})$ be the signature of the language of arithmetic with one name added, `c`. For any number n, let A_n be the sentence

$$\texttt{¬(c = }\langle n\rangle\texttt{)}$$

Let $X_n = \{A_0, A_1, \ldots, A_n\}$. Prove that for any number n, the set $\mathsf{PA} \cup X_n$ is consistent.

Problem 7

For each of the following, either give an example or explain why there is no example.

(a) A theory that is sufficiently strong, effectively axiomatizable, and decidable.

(b) A structure S such that $\mathrm{Th}\, S$ (the theory of S) is sufficiently strong and semi-decidable.

(c) A definitional expansion of $\mathbb{S}$ which has an effectively axiomatizable theory $\mathrm{Th}\, S$.

(d) A theory T such that T includes all the theorems of Peano arithmetic (PA), T is consistent, and T contains at least one theorem that is false in the standard model of arithmetic $\mathbb{N}$.