24.118: Paradox and Infinity, Spring 2019
Problem Set 10: Gödel's Theorem

How these problems will be graded:

- In Part I there is no need to justify your answers. Assessment will be based on whether your answers are correct.
- In Part II you must justify your answers. Assessment will be based both on whether you give the correct answer and on how your answers are justified. (In some problem sets I will ask you to answer questions that don't have clear answers. In those cases, assessment will be based entirely on your justification. Even if it is unclear whether your answer is correct, it should be clear whether or not the reasons you have given in support of your answer are good ones.)
- *No answer may consist of more than 150 words.* Longer answers will not be given credit. (Showing your work in a calculation, a proof, or a computer program does not count towards the word limit.)
- You may consult published literature and the web. You must, however, credit all sources. Failure to do so constitutes plagiarism and can have serious consequences. For advice about when and how to credit sources see: https://integrity.mit.edu/ (You do not need to credit course materials.)

Note: Several of the questions below pertain to a specific formal language L, which is defined in Section 10.2 of the textbook. The symbol "$\mathcal{L}$", in contrast, is used as a variable to range over arbitrary formal languages.

Part I

1. The following are meant to test your ability to formalize sentences of L. You may make use of any of the notational abbreviations introduced in chapter 10 of the textbook (or its appendix).

   (a) Define an expression "Mult17(x_i)" of L that is true if and only if x_i is a multiple of 17. (5 points)

   (b) Define an expression "PerfectSquare(x_i)" of L that is true if and only if x_i is a perfect square. (5 points)

   (c) Define an expression "IrrationalRoot(x_i)" of L that is true if and only if $\sqrt{x_i}$ is an irrational number. (5 points)

(d) *Extra Credit:* Define an expression "Perfect(x_i)" of L that is true if and only if x_i is a perfect number.[1] (5 points)

(e) Find a sentence of L that expresses the (incorrect) claim that the natural numbers are dense: between any two natural numbers there is a third. (5 points)

(f) Find a sentence of L that expresses Fermat's Last Theorem: there are no positive integers a, b, c and $n > 2$ such that $a^n + b^n = c^n$. (5 points)

(g) Find a sentence of L that expresses Euclid's Lemma: if a prime p divides the product of natural numbers a and b, then p must divide at least one of a and b. (5 points)

Part II

2. This question has three parts: (5 points each)

   (a) Is it possible to construct a Turing Machine M_1 that runs forever outputting sentences of L in such a way that every arithmetical truth is eventually output by M_1?

   (b) Is it possible to construct a Turing Machine M_2 that runs forever outputting sentences of L in such a way that no arithmetical falsehood is ever output by M_2?

   (c) Is it possible to construct a Turing Machine that satisfies both the conditions of M_1 above and the conditions of M_2 above?

3. Let $L^{10^{100}}$ be the sublanguage of L consisting of formulas of L with 10^{100} symbols or fewer. Is $L^{10^{100}}$ immune from Gödel's Theorem? More specifically, is there a Turing Machine $M^{10^{100}}$ that when run on an empty input eventually outputs every true sentence of $L^{10^{100}}$ without ever outputting a false sentence of $L^{10^{100}}$?

   If you think $M^{10^{100}}$ exists, give a sketch of how it might work. If you think $M^{10^{100}}$ cannot exist, adapt the proof of Gödel's Theorem in section 10.3 of the textbook to show that its existence would lead to contradiction. (10 points.)

4. The following definitions are adapted from sections 10.3.2 and 10.3.3 of the textbook:

   An **axiomatization** for a language $\mathcal{L}$ is a system for proving claims within $\mathcal{L}$. It consists of two different components, a set of axioms and a set of rules of inference:

   - An **axiom** is a sentence of $\mathcal{L}$ that one treats as provable by fiat.

     When $\mathcal{L}$ is the language of arithmetic, for example, is is natural to select "$0 = 0$" as an axiom.

[1] Recall that a perfect number is a number that is equal to the sum of its proper divisors. (A proper divisor of n is a divisor of n distinct from n.)

- A **rule of inference** is a rule that allows you to count a sentence of $\mathcal{L}$ as provable, given that other sentences of $\mathcal{L}$ are provable.

  An example of a rule of inference is *modus ponens*, which says that ψ is provable, given that ϕ and "if ϕ, then ψ" are both provable.

For an axiomatization of $\mathcal{L}$ to be **complete** is for every true sentence of $\mathcal{L}$ to be provable on the basis of that axiomatization. For it to be **consistent** is for it to never be the case that both a sentence of $\mathcal{L}$ and its negation are provable on the basis of that axiomatization.

Now suppose that $\mathcal{L}$ consists of the following sentences:

- "pigeons can't fly" and "camels are mammals" are both sentences of $\mathcal{L}$.
- "it is not the case that pigeons can't fly" and "it is not the case that camels are mammals" are both sentences of $\mathcal{L}$.
- If ϕ and ψ are sentences of $\mathcal{L}$, then so is "it is both the case that ϕ and that ψ".
- Nothing else is a sentence of $\mathcal{L}$.

(a) Show that the following is an inconsistent axiomatization of $\mathcal{L}$: (5 points)

Axioms: "camels are mammals"

Rules of Inference: (1) If ϕ is provable, then so is any sentence of the form "it is both the case that ϕ and that ψ". (2) If "it is both the case that ϕ and that ψ" is provable, then ϕ and ψ are both provable.

(b) Show that the axiomatization of problem (4a) is complete. (5 points)

(c) Specify a consistent and complete axiomatization of $\mathcal{L}$. (5 points; no need to justify your answer.)

5. The textbook describes a proof of Gödel's Theorem that is based on the following lemma:

Lemma 1 L contains a formula (abbreviated "Halt(x_i)") which is true of a number k if and only if the kth Turing Machine halts on input k.

As it happens, the following lemma is also true:

Lemma 2 L contains a formula (abbreviated "BB(x_i, x_j)") which expresses the busy beaver function. In other words: BB(n, k) if and only if the productivity of the most productive Turing Machine with n states or fewer is k.

Prove Gödel's Theorem by relying on Lemma 2 rather than Lemma 1. (30 points)

(*Hint:* You may assume that the Busy Beaver Function is not computable.)

Remember that the word limit doesn't apply to proofs.