CS 4814: Homework 2

due 09/09, 11:59pm

Each problem is worth 20 points, except the warm-up problem (see below).

Warm Up

You do not have to turn anything in for this problem. However, if you are struggling with the problems below, we recommend solving this one. It is a bit easier, but the proof has a similar flavor to the ones below and will help you grow your diagonalization muscles.

In class, we proved the existence of undecidable languages using the following fact:

> The set of binary strings $\{0,1\}^*$ has *strictly smaller cardinality* than the set of all languages $2^{\{0,1\}^*} = \{S \mid S \subseteq \{0,1\}^*\}$, that is, for every function $f\colon \{0,1\}^* \to 2^{\{0,1\}^*}$ there exists some language $S_0 \subseteq \{0,1\}^*$ not in the image of f.

Prove this fact using diagonalization.

Question 1

Show that the following language B is undecidable:

$$B = \{\langle M \rangle \,:\, M \text{ is a machine which runs in } n^3 + 100 \text{ time}\}.$$

Question 2

In this problem, you will complete the proof of the time hierarchy theorem from lecture. Recall that our goal is to show that $\mathrm{TIME}(n^2)$ is strictly contained in $\mathrm{TIME}(n^4)$. We gave a language L and proved the following statements:

- No n^3-time TM decides L.
- $L \in \mathrm{TIME}(n^4)$.

To complete the proof, we need to show something a bit different than (1)—that $L \notin \mathrm{TIME}(n^2)$. (Here, $\mathrm{TIME}(n^k)$ contains all problems solved by $O(n^k)$-time Turing machines.)

Recall that

$$L = \{\langle M \rangle :\, M(\langle M \rangle) \text{ outputs } 0 \text{ in at most } n^3 \text{ steps}\}.$$

(a) Adapt the proof that no n^3-time TM decides L to show that every n^3-time TM fails to solve L on infinitely many inputs.

(b) Show that for every language $A \in \mathrm{TIME}(n^2)$, there exists an n^3-time Turing machine that solves A on all but a finite number of inputs.

(c) Use part (a) and (b) to conclude that $L \notin \mathrm{TIME}(n^2)$.

Question 3

In class, we sketched a proof of Gödel's incompleteness theorem, saying that any proof system for first order logic over the natural numbers is either unsound or incomplete. This does not, however, yield an *explicit* formula in first order logic which is true but unprovable in a given proof system. In this problem you will find such a formula.

We will work, as before, in first-order logic over the natural numbers. This is first-order logic (so it has boolean operations and existential and universal quantification) augmented with multiplication, addition, and equality. Recall that a proof system for this theory is a TM V (also called a *verifier*) which always halts. The machine V interprets its input as $\langle \phi, \pi \rangle$, where ϕ is a formula in the logic (and ϕ has no free variables) and π is a (candidate) *proof* for ϕ. The proof system V is sound if for every false ϕ there is no π so that $V(\langle \phi, \pi \rangle) = 1$, and it is complete if for every true ϕ there exists π so that $V(\langle \phi, \pi \rangle) = 1$.

For the rest of the problem, fix a sound verifier V. Consider a fixed enumeration $\phi_1, \phi_2, \ldots$ of formulas with one free variable, and a fixed enumeration $\pi_1, \pi_2, \ldots$ of (potential) proofs π. You may assume that our logic contains a formula ψ with one free variable x such that $\psi(x)$ is true if and only if there is y so that $V(\neg\phi_x(x), \pi_y) = 1$. (In words, $\psi(x)$ is true if and only if the formula $\neg\phi_x(x)$ is provable within the proof system given by V.) Using ψ, find an explicit sentence of the logic which is true but not provable by V.