Theory of Computation, Fall 2023
Assignment 9 Solutions

Q1. Define $g : \mathcal{N} \times \mathcal{N} \to \mathcal{N}$ to be

$$g(m, n) = f(f(\dots f(n) \dots)),$$

where there are m compositions. g can also be written as follows.

$$g(0, n) = f(n)$$
$$g(m+1, n) = f(g(m, n))$$

Since f is primitive recursive, so is g.

We have that $F(n) = g(n, n)$. That is,

$$F(n) = g(id_{1,1}(n), id_{1,1}(n)).$$

F is the composition of primitive recursive functions. Therefore, F is primitive recursive.

Q2. Fix an arbitrary $k \geq 2$. For $i \in [1, k]$, define P_i as follows.

$$P_i(n_1, \dots, n_k) = \begin{cases} 1, & \text{if } (n_i = \max\{n_1, \dots, n_k\}) \wedge (\forall j < i, n_j \neq \max\{n_1, \dots, n_k\}) \\ 0, & \text{otherwise} \end{cases}$$

P_i is a primitive recursive predicate since P_i can also be written as

$$P_i(n_1, \dots, n_k) = (n_i > n_1) \wedge \dots \wedge (n_i > n_{i-1}) \wedge (n_i \geq n_{i+1}) \wedge \dots \wedge (n_i \geq n_k)$$

Note that

$$\varphi_k(n_1, \dots, n_k) = \sum_{i=1}^{k} P_i(n_1, \dots, n_k) \cdot n_i$$

That is, φ_k is a composition of primitive recursive functions. Thus φ_k is primitive recursive.

Q3. Since $A \in \mathcal{P}$, A is decided by some deterministic Turing machine M_A with polynomial running time.

Construct a deterministic Turing machine $M_{\overline{A}}$ as follows.

$M_{\overline{A}}$ = on input w:
1. Run M_A on w
2. If M_A accepts w
3. Reject w
4. Else (M_A rejects w)
5. Accept w

It is easy to see that $M_{\overline{A}}$ decides $\overline{A}$ in polynomial time. Therefore, $\overline{A} \in \mathcal{P}$.

Q4. By the conclusion of Q3, we know that $A \in \mathcal{P}$ implies that $\overline{A} \in P$. Since $\mathcal{P} \subseteq \mathcal{NP}$, we have $A \in \mathcal{NP}$ and $\overline{A} \in \mathcal{NP}$. Therefore, $A \in \mathcal{NP} \cap$ co-$\mathcal{NP}$.

Q5. The following Turing machine V is a polynomial-time verifier for L.

V = on input "G""p":
1. If p does not represent a cycle in G
2. reject
3. traverse along p
4. accept if p visit every vertex of G exactly once, and reject otherwise