Exercise 1: Let **#CLIQUE** be the problem of counting how many k-cliques exist in a given graph for a given positive integer k. Show that **#CLIQUE** $\in \mathbf{P}^{\#\mathbf{SAT}}$.

Solution: I proceed by direct reduction. Let $G = (V, E)$ be an undirected graph with n vertices and $1 \leq k \leq n$ an integer. I now claim that the number N' of *ordered* k-cliques can be found in polynomial time with access to a **#SAT** oracle. This is enough because given the number of ordered k-cliques is $N' = k!N$ and $N' \mapsto N'/k! \in \mathbf{FP}$.

Let $T = (u_1, \dots u_k) \in V^k$ be a k-tuple of vertices. The elements of T form a k-clique in G if and only if $\{u_i, u_j\} \in E$ for all $1 \leq i < j \leq k$ (note that this implies that all the elements are different, since I am assuming that G is a simple graph, with no loops). To encode this information into a boolean formula I consider the variables x_i^u (for $u \in V$ and $i \in [k]$), representing whether $u_i = u$. Define

$$F \coloneqq \{(u, w) \in V^2 \mid \{u, w\} \notin E\}.$$

Note that this includes the diagonal entries (u, u). The clique condition can then be expressed as

$$\mathcal{C} \coloneqq \bigwedge_{(u,w) \in F, (i,j) \in \binom{[k]}{2}} (\neg x_i^u \vee \neg x_j^w),$$

which is in CNF. An assignment $V \times [k] \to \{0, 1\}$ represents a k-tuple of vertices if and only if exactly one vertex is selected for each entry of T. I do this in two steps. First I introduce a formula that ensures *at most* one vertex is selected for each entry:

$$\mathcal{B} \coloneqq \bigwedge_{\{u,w\} \in \binom{V}{2},\, i \in [k]} (\neg x_i^u \vee \neg x_i^w).$$

Then, I introduce the following other formula, which ensures *at least* one vertex is selected for each entry.

$$\mathcal{A} \coloneqq \bigwedge_{i \in [k]} \left(\bigvee_{u \in V} x_i^u \right).$$

Formulas $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ can clearly be obtained from G (represented, for example, as an adjacency matrix) in polynomial time. Furthermore, there is a bijection between assignments $V \times [k] \to \{0, 1\}$ satisfying $\mathcal{A} \wedge \mathcal{B} \wedge \mathcal{C}$ and ordered k-cliques in G. Therefore, a single query to a **#SAT** oracle is enough to count the number N' of ordered k-cliques in G in polynomial time. Then computing $N = N'/k!$ yields the actual number of k-cliques.

Exercise 2: We know that $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}}$ (Toda's theorem). What would be the consequences of the reverse inclusion $\mathbf{P}^{\#\mathbf{P}} \subseteq \mathbf{PH}$?

Solution: Suppose that $\mathbf{P}^{\#\mathbf{P}} \subseteq \mathbf{PH}$. I will argue that the polynomial hierarchy collapses to a finite level. consider the language

$$L = \{\langle \mathcal{F}, k\rangle \mid \mathcal{F} \text{ is a formula in CNF with less than } k \text{ solutions}\}.$$

Clearly, $L \in \mathbf{P}^{\#\mathbf{P}}$, since given $\langle \mathcal{F}, k\rangle$ one can count the number of solutions to $\mathcal{F}$ with a **#SAT** oracle call and then check whether this number is less than k, all in polynomial time. By assumption, $L \in \mathbf{PH}$ and in particular $L \in \mathbf{\Sigma}_i^p$ for some $i \geq 0$. Consider now another language $A \in \mathbf{PH}$. By Toda's theorem, $A \in \mathbf{P}^{\#\mathbf{P}} = \mathbf{P}^{\#\mathbf{SAT}}$. Suppose that a machine M decides A in polynomial time with access to a **#SAT** oracle. I now argue that each call to the **#SAT** oracle can be replaced by a polynomial-time computation with access to an oracle for L, obtaining a machine M' that decides A in polynomial time with access to an oracle for L. Therefore, $A \in \mathbf{P}^L \subseteq \mathbf{P}^{\mathbf{\Sigma}_i^p} = \mathbf{\Delta}_{i+1}^p$, proving that $\mathbf{PH} = \mathbf{\Delta}_{i+1}^p$.

Let us now see the promised reduction. Suppose the computation of M on input x takes $p(|x|)$ steps, where p is a polynomial. The number of variables in the formula $\mathcal{F}$ of an oracle call in such a computation is at most $|\mathcal{F}| \leq |x| + p(|x|) = p'(|x|)$, since the formula has to be written on the tape of the machine. The number of solutions to $\mathcal{F}$ is at most $2^{p'(|x|)}$. The exact number of solutions to $\mathcal{F}$ can be computed by running binary search on the range $[0, 2^{p'(|x|)}]$, taking $\mathcal{O}(\log(2^{p'(|x|)})) = \mathcal{O}(p'(|x|))$ iterations and making a call to the oracle for L in each iteration. The number of steps of each iteration is in the order of the size of the integers that are being compared and operated on, which is also $\mathcal{O}\left(\log(2^{p'(|x|)})\right) = \mathcal{O}\left(p'(|x|)\right)$. All in all, the call to the oracle for **#SAT** can be replaced by a computation with access to an oracle for L that takes $\mathcal{O}\left(p'(|x|)^2\right)$ steps, and M' runs in time $\mathcal{O}\left(p(|x|)p'(|x|)^2\right)$, justifying the claim that $A \in \mathbf{P}^L$.