1. (Courtroom complexity.) In this problem we study a slightly peculiar complexity class that we’ll call $S_2^P$. Informally, we say $L \in S_2^P$ whenever the following circumstances hold.

There are two lawyers, Yolanda and Zeyuan, whose job is to argue in front of judge Victor about whether or not $x \in L$. Whenever $x \in L$, there is something Yolanda can say that will convince judge Victor that indeed $x \in L$, no matter what Zeyuan says. Conversely, whenever $x \not\in L$, there is something Zeyuan can say that will convince judge Victor that $x \not\in L$, no matter what Yolanda says.

More precisely, we say that $L \in S_2^P$ if there is a polynomial $p(n)$ and a polynomial-time algorithm $V$ such that

\[ x \in L \Rightarrow \exists y \forall z \; V(x, y, z) = 1, \]
\[ x \not\in L \Rightarrow \exists z \forall y \; V(x, y, z) = 0. \]

(Recall “$\exists y$ with $|y| \leq p(|x|)$”, etc.)

(a) Show that $S_2^P$ is closed under complement: $\text{co}S_2^P = S_2^P$.

(b) Show that $S_2^P \subseteq \Sigma_2^P \cap \Pi_2^P$.

(c) Show $\text{NP} \subseteq \text{P}/\text{poly} = \Rightarrow \text{PH} = S_2^P$. (This is an improvement on the Karp–Lipton Theorem, by part (b)... but in fact, you can solve this problem by almost literally repeating the proof of Karp–Lipton.)

(d) Show that $\text{P}^\text{NP} \subseteq S_2^P$.

2. (A route to $P \neq \text{NP}$?) Let $c_n$ denote the maximum number of gates needed by a Boolean circuit to compute any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Shannon and Lupanov showed that $c_n \approx 2^n / n$, but we will be interested in the literal exact value of $c_n$. Let us say that a language $L$ has maximal circuit complexity if $L \cap \{0, 1\}^n$ requires circuits of size $c_n$ for every $n$.

Show that if every language in $\text{E}$ has non-maximal circuit complexity (i.e., just one gate can be saved somewhere in the circuit family) then $P \neq \text{NP}$. (Recall that $E = \bigcup_c \text{TIME}(2^{cn})$.)

3. (Limited SAT queries.) When $C$ is a complexity class, the notation $C^{A[k]}$ means the same class where at most $k$ oracle queries to the language $A$ are allowed. As usual, $C^{\text{NP}[k]}$ denotes the union of $C^{A[k]}$ over all $A \in \text{NP}$; equivalently, it’s $C^{\text{SAT}[k]}$. In studying the Polynomial Time Hierarchy, we observed that when $C = \text{NP}$, we could massively reduce the number of queries used: $\text{NP}^{\text{NP}} = \text{NP}^{\text{NP}[\text{poly}(n)]} = \text{NP}^{\text{NP}[1]}$. The same is (seemingly) not true when $C = \text{P}$; it is believed that $\text{P}^{\text{NP}[1]} \subset \text{P}^{\text{NP}[2]} \subset \text{P}^{\text{NP}[3]} \subset \cdots$.

In this problem, we will look at an interesting class: $\text{P}^{\text{NP}[\text{log}]}$, which is short for $\text{P}^{\text{NP}[O(\text{log} n)]}$, the class of languages decidable in polynomial time by a SAT-oracle Turing Machine that makes at most $O(\text{log} n)$ oracle queries on inputs of length $n$.

(a) Show that the following two problems are in $\text{P}^{\text{NP}[\text{log}]}$: UNIQUE-MAX-CLIQUE, the language of all graphs whose largest clique is unique; ODD-MAX-CNF-SAT, the language of all CNF formulas for which the maximum number of clauses that can be satisfied by any truth assignment is odd.
(b) Define $P_{\parallel}^{NP[r]}$ to be the class of all languages decidable in polynomial time by a SAT-oracle Turing Machine that makes at most $r$ nonadaptive oracle queries. This means that the machine can only interact with “the oracle” one time, in the following way: it can submit $r$ separate oracle queries, and get back the $r$ answers. Show that $P_{\parallel}^{NP[k]} \subseteq P_{\parallel}^{NP[2^k-1]}$, even for $k = O(\log n)$, and hence $P_{\parallel}^{NP[\log]} \subseteq P_{\parallel}^{NP}$.

(c) Building on work of Gilbert, Michael Fischer showed the following result: For every $n$, there is an $n$-input, $n$-output Boolean circuit, consisting of poly($n$) AND gates, poly($n$) OR gates, and $\lceil \log_2(n+1) \rceil$ NOT gates, such that on input $(x_1, x_2, \ldots, x_n)$, the output is $(\neg x_1, \neg x_2, \ldots, \neg x_n)$.\(^1\) If you have never seen this before, I very strongly urge you to try to prove this result in the case $n = 3$; it’s a great puzzle! But anyway, you can assume Fischer’s result.

Show an almost-opposite containment to part (b): $P_{\parallel}^{NP[2^k-1]} \subseteq P_{\parallel}^{NP[k+1]}$, even for $k = O(\log n)$, and hence $P_{\parallel}^{NP[\log]} = P_{\parallel}^{NP}$.

(0-point bonus problem: Can you get the exact-opposite containment, $P_{\parallel}^{NP[2^k-1]} \subseteq P_{\parallel}^{NP[k]}$ in case $k = 2$? Can you get it in general?)

\(^1\) Also, the construction is $P$-uniform.