A strong nondeterministic Turing Machine has, in addition to its $q_{\text{accept}}$ and $q_{\text{reject}}$ states, a special state $q^*$. Such a Turing Machine accepts its input if all computation paths lead to $q_{\text{accept}}$ and $q^*$ states, and it rejects its input if all computation paths lead to $q_{\text{reject}}$ and $q^*$ states. Moreover, on every input, there is at least one computation path leading to $q_{\text{accept}}$ or $q_{\text{reject}}$. Show that the class of languages decided by a strong nondeterministic Turing Machine in polynomial time is exactly $\text{NP} \cap \text{coNP}$.

In this problem you will prove Mahaney’s Theorem: a sparse language $S$ cannot be $\text{NP}$-complete unless $\text{P} = \text{NP}$. Throughout this problem, $S$ is a sparse language in $\text{NP}$ with a polynomial bound $p(n)$ on the number of strings of length at most $n$.

(a) Explain briefly where the proof of the special case of Mahaney’s Theorem for unary languages (from class) breaks down for sparse languages.

(b) Show that if $\text{SAT}$ reduces to $S$ in polynomial time via reduction $R$, then a procedure very similar to the one for unary languages from class decides $\text{SAT}$ in polynomial time, and hence implies $\text{P} = \text{NP}$.

(c) Define $c(n)$ to be the exact number of strings of length at most $n$ in $S$ (clearly $c(n) \leq p(n)$ for all $n$). Argue that the following language is in $\text{NP}$:

$$\hat{S} = \{(x, 1^k) : k < c(|x|) \text{ or } (k = c(|x|) \text{ and } x \notin S)\}.$$

Hint: do not try to compute $c(|x|)$; rather, focus on describing an $\text{NP}$ algorithm that decides $\hat{S}$ properly under the assumption that $k = c(|x|)$, and then see what your algorithm does when $k \neq c(|x|)$.

(d) Finally we assume $S$ is $\text{NP}$-complete. Thus, everything in $\text{NP}$ reduces to $S$, and we give names to two of these reductions: let $T$ be a polynomial-time reduction from $\text{SAT}$.

---

1A language is decided (as usual) if every input is either accepted or rejected according to the accept/reject criteria for this type of machine.
to \(S\), and let \(U\) be a polynomial-time reduction from \(\hat{S}\) to \(S\). Using \(T\) and \(U\), describe a family of “candidate reductions from \(\overline{\text{SAT}}\) to \(S\),” \(R_k\), with the following properties:

\[
R_k(\phi) \in S \quad \text{if} \quad k < c(|T(\phi)|)
\]

\[
R_k(\phi) \in S \iff \phi \in \overline{\text{SAT}} \quad \text{if} \quad k = c(|T(\phi)|)
\]

\[
R_k(\phi) \not\in S \quad \text{if} \quad k > c(|T(\phi)|)
\]

(e) Using parts (b) and (d), prove Mahaney’s Theorem. You may need to modify part (b) slightly so that on a given input \(\phi\), the procedure only applies \(R\) to formulae \(\phi'\) for which \(|\phi'| = |\phi|\). This should require at most a syntactic change: we can think of any partial assignment of values to variables in \(\phi\) as having the same length as \(\phi\) if we don’t perform any simplification. To solve a similar technical problem, you may want to prove and make use of the following lemma:

**Lemma 2.1** If language \(L \subseteq \Sigma^*\) is \(\text{NP}\)-complete, then language \(L' \in (\Sigma \cup \{\#\})^*\) defined by

\[
L' = \{x\#^i : x \in L, i \geq 0\}
\]

is \(\text{NP}\)-complete. If \(L\) is sparse then \(L'\) is sparse.

(f) Argue that if \(\text{P} = \text{NP}\), then there are sparse \(\text{NP}\)-complete languages (under polynomial-time, many-one reducibility).

3. A directed graph \(G = (V, E)\) is **strongly connected** if for every pair of vertices \((x, y)\) there is a directed path from \(x\) to \(y\) and a directed path from \(y\) to \(x\). Consider STRONGLY CONNECTED, the language of graphs \(G\) that are strongly connected. Either show that this problem is in \(L\), or prove a complexity consequence of such a containment.