For problems that involve nondeterministic complexity classes, the solutions maybe simpler when phrased in terms of “certificates” (instead of non-determinism).

Problem 1:

(a) Let $L_1, L_2$ be languages in $\text{NP}$. Are $L_1 \cup L_2$ and $L_1 \cap L_2$ necessarily in $\text{NP}$?

(b) Let $L_1, L_2$ be languages in $\text{NP}$. Show that $L_1 L_2$ and $L_1^*$ are in $\text{NP}$.

(c) Let $L_1, L_2$ be languages in $\text{P}$. Show that $L_1 L_2$ and $L_1^*$ are in $\text{P}$.

(d) Let $L_1, L_2$ be languages in $\text{NP} \cap \text{co-NP}$. Show that their symmetric difference

$$L_1 \oplus L_2 \overset{\text{def}}{=} \{x \mid x \text{ is in exactly one of } L_1, L_2\}$$

is also in $\text{NP} \cap \text{co-NP}$.

Problem 2:

(a) Show that the halting problem is $\text{NP}$-hard. Is it $\text{NP}$-complete?

(The halting problem is given by the language $H = \{(\langle M \rangle, x) \mid M \text{ is a TM that halts on input } x\}$. You may recall that $H$ is undecidable.)

(b) Show that $\overline{\text{SAT}}$ (the complement of SAT) is $\text{NP}$-hard under Cook reductions. That is, every language in $\text{NP}$ reduces to $\overline{\text{SAT}}$ via a Cook reduction. (On the other hand, we believe $\overline{\text{SAT}}$ is not $\text{NP}$-hard (under Karp reductions). If it were, then $\text{NP} = \text{co-NP}$.)

Problem 3:

Show that the following two statements are equivalent (we don’t know if they are true):

(a) Every unary language in $\text{NP}$ is also in $\text{P}$.

(b) $\text{DTIME}(2^{O(n)}) = \text{NTIME}(2^{O(n)})$ (these classes are called $\text{E}$ and $\text{NE}$, respectively).

Hint: It takes $\Theta(\log n)$ bits to encode the number “$n$” in binary.

Problem 4:

Give a parsimonious Karp reduction from SAT to 3SAT.

---

1 A language is unary if it is a subset of $\{1\}^*$ — that is, it only uses one symbol of the alphabet.
Problem 5:
In this problem, we analyze a reduction from 3SAT to the following language:

\[
\text{MAX-2SAT} = \{ (\phi, k) \mid \phi \text{ is a 2-CNF formula, and there is an assignment that satisfies at least } k \text{ clauses} \}
\]

Our reduction is the following: Given a 3SAT instance \(\phi\), we will output a MAX-2SAT instance \((\phi', k)\), where \(\phi'\) is a 2-CNF formula. To construct \(\phi'\), do the following: for each clause \((x \lor y \lor z)\) in \(\phi\), add the following 10 clauses to \(\phi'\) (where \(w\) is a fresh variable for each clause):

\[
(x), (y), (z), (\neg x \lor \neg y), (\neg y \lor \neg z), (\neg x \lor \neg z), (w), (x \lor \neg w), (y \lor \neg w), (z \lor \neg w)
\]

Find a value of \(k\) such that \((\phi', k) \in \text{MAX-2SAT}\) if and only if \(\phi \in 3\text{SAT}\). Prove the correctness of the reduction.

Problem 6 (Extra credit):
Show that 2SAT is in P.

Hint: Consider a directed graph with all the literals as nodes, and edges as implications \((x \lor y)\) corresponds to \((\neg x \Rightarrow y)\) and \((\neg y \Rightarrow x)\). Look to derive contradictions of the form \((\neg x \Rightarrow x)\) and \((x \Rightarrow \neg x)\). What do such contradictions tell you about a possible satisfying assignment?

Problem 7 (Extra credit) [See Arora-Barak (web-draft) Chapter 2, Exercise #13]:
Show that if there is a unary language that is \(\text{NP}\)-complete, then \(P = \text{NP}\).

Problem 8 (Extra credit):
Consider the following language:

\[
\text{MAX-CUT} = \{ (G, k) \mid G \text{ is a multigraph with a cut of size at least } k \}
\]

A cut in a graph is a partition of its vertices into two parts. The size of the cut is the number of edges which “cross” the cut (whose endpoints are in opposite parts). A multigraph means we allow duplicate edges.

We now analyze a reduction from MAX-2SAT to MAX-CUT. Given an instance \((\phi, k)\) of MAX-2SAT, let \(n\) be the number of variables occurring in \(\phi\), and \(m\) the number of clauses. Consider the following graph:

\(G_{\phi}\) is a graph with vertices labeled \(x_i\) and \(\neg x_i\) for each variable \(x\) occurring in \(\phi\), and two special vertices labeled \(T\) and \(F\). We add \(5m\) edges between \(T\) and \(F\), and \(5m\) edges between each pair \((x_i, \neg x_i)\) — see Figure 1. Then, for each clause \((x \lor y) \in \phi\), where \(x\) and \(y\) are literals, we add the following 7 edges (see Figure 2):

- \((x, y)\), \((T, x)\), \((T, y)\).
- Two copies of the edges \((x, F)\) and \((y, F)\).

(a) Show that in the largest cut in \(G_{\phi}\), \(T\) and \(F\) must be in opposite parts.

(b) Show that in the largest cut in \(G_{\phi}\), the vertices corresponding to \(x\) and \(\neg x\) must be in opposite parts.

(c) Argue that \((\phi, k) \in \text{MAX-2SAT}\) if and only if \((G_{\phi}, 5m + 5mn + 4k + 2(m - k)) \in \text{MAX-CUT}\).
Figure 1: Starting graph for $G_\phi$, where $\phi$ has $n$ variables, $x_1, \ldots, x_n$.

Figure 2: Edges to add for a clause of the form $(x \lor y)$