In many solutions, we use the following definition: For any language $A$ and function $f : \mathbb{N} \to \mathbb{N}$, let $A_f(n) = \{(x, 1^{f(|x|)}) \mid x \in A\}$. Let $pad_f(n)(x) = (x, 1^{f(|x|)})$. Then

$$x \in A \iff pad_f(n)(x) \in A_f(n)$$

and $pad_f(n)$ is computable in polynomial time, so $A \leq^p A_f(n)$.

**Problem 1**

Recall that $\text{NEXP}$ is defined by

$$\text{NEXP} = \bigcup_{c \geq 1} \text{NTIME}(2^{nc}).$$

Give a definition of $\text{NEXP}$ that does not involve non-deterministic Turing machines, analogous to the verifier definition of $\text{NP}$ seen in class, and prove that your definition is equivalent to the above definition using non-deterministic Turing machines.

**Solution to Problem 1**

Define $\text{NEXP}_1 = \bigcup_{c \geq 1} \text{NTIME}(2^{nc})$.

Define the class $\text{NEXP}_2$ as follows: A language $L \in \text{NEXP}_2$ if there exists a relation $R(x, y)$ decidable in $p(|x|) \in O\left(2^{nc}\right)$ for some $c \in \mathbb{N}_+$ such that

$$x \in L \iff \exists y, |y| \leq p(|x|) \land (x, y) \in R.$$

We'll show that $\text{NEXP}_1 = \text{NEXP}_2$.

Consider an arbitrary language $L \in \text{NEXP}_1$. Let $M$ be a NDTM deciding $L$ running in time $q(|x|) \in O\left(2^{nc}\right)$ for some $c \in \mathbb{N}_+$. We'll define the relation $R(x, y)$ by

$$R = \{(x, y) \mid y \text{ is a sequence of non-deterministic choices for an accepting path of } M \text{ on input } x\}.$$
We observe that $R(x, y)$ is decidable in $q(|x|)$ time by exhibiting a DTM deciding $R(x, y)$ with the required running time. The desired TM $N$ simply simulates $M$ on $x$, deterministically choosing a single branch of $M$’s computation at the $i$th step of $M$ according to the $i$th bit of $y$. $N$ runs in $\text{poly}(q(|x|)) \in O\left(2^{\text{poly}(|x|)}\right)$ time since any valid $y$ has length $q(|x|)$ and looking up the $i$th bit of $y$ between executing successive steps of $M$ incurs only a polynomial slowdown over executing $M$. Furthermore, if $x \in L$ there is some accepting computation of $M$ on input $x$, and if $x \notin L$ then all computations of $M$ on input $x$ are rejecting, so we obtain

$$x \in L \iff \exists y, |y| \leq q(|x|) \land (x, y) \in R.$$  

We conclude that $L \in \text{NEXP}_2$. Thus, $\text{NEXP}_1 \subseteq \text{NEXP}_2$.

Now consider an aritrary language $L \in \text{NEXP}_2$. Let $R$ be the relation required by the definition of $\text{NEXP}_2$, and let $M$ be a DTM deciding $R(x, y)$ in $p(|x|) \in O\left(2^{\text{poly}(|x|)}\right)$ time for some $c \geq 1$.

We’ll construct a NDTM $N$ deciding $L$ as follows: Given input $x$, $N$ will non-deterministically write down a string $y$ of length $p(|x|)$ on the input tape following $x$, and then will simulate $M$ on input $(x, y)$, accepting $x$ on a given path iff $M(x, y)$ accepts. By construction, $N$ runs in $O(p(|x|))$ time, since simulating $M$ takes $p(|x|)$ time and writing $y$ takes $p(|x|)$ time. Furthermore $x \in L$ if and only if there is some $y$ such that $M(x, y)$ accepts, so $N$ will accept if and only if $x \in L$. Thus we have $L \in \text{NEXP}_1$.

We conclude that $\text{NEXP}_2 \subseteq \text{NEXP}_1$.

Finally, we conclude that $\text{NEXP}_1 = \text{NEXP}_2$.

**Problem 2**

Recall that $\text{E} = \text{DTIME}(2^{O(n)})$ is the class of problems solvable by a deterministic Turing machine in time $2^{O(n)}$, where $n$ is the length of the input. We say that a language $A$ has a many-to-one polynomial time reduction to a language $B$, written $A \leq^p_m B$ if there is a polynomial time computable function $f(\cdot)$ such that for every instance $x$ we have $x \in A \iff f(x) \in B$.

- Show that $\text{NP}$ is closed under polynomial many-to-one reductions, that is $A \leq^p_m B$ and $B \in \text{NP}$ implies $A \in \text{NP}$.

- Show that if $\text{E}$ were closed under many-to-one reductions, we would have a contradiction to the time hierarchy theorem. Conclude that $\text{NP} \neq \text{E}$.

**Solution to Problem 2**

1. Let $A$ and $B$ be languages with $B \in \text{NP}$ and $A \leq^p_m B$. Then there exists a NDTM $M$ that decides $B$ in time $\text{poly}(|x|)$. By the definition of a many-to-one polynomial
reduction, there exists a polytime computable function \( f(\cdot) \) such that \( x \in A \iff f(x) \in B \). Consider the following algorithm for deciding \( A \):

Compute \( f(x) \) in polytime. Then run \( M \) on \( f(x) \), accepting and rejecting according to \( M(f(x)) \). Trivially \( M(f(x)) = 1 \iff x \in A \). Furthermore, \( |f(x)| = O(\text{poly}(|x|)) \) and \( \text{poly}(\text{poly}(|x|)) = \text{poly}(|x|) \), so running \( M \) on \( f(x) \) takes polynomial time. Thus we have a polynomial time algorithm on a NDTM to decide \( A \) and we conclude that \( A \in \text{NP} \).

2. Assume \( \text{E} \) is closed under polynomial many-to-one reductions. Then let \( A \in \text{EXP} \) be an arbitrary language in \( \text{EXP} \). Then there exists a TM \( M \) that decides \( A \) in time \( O(2^{O(n^c)}) \) for some \( c \in \mathbb{N} \). Consider the padded language \( A_{n^2} \). As stated above, \( A \leq_m A_{n^2} \). Trivially, \( A_{n^2} \in \text{E} \) since we can decide \( y = (x, 1^{|x|^f}) \in A_{n^2} \) in time \( O(2^{2|y|}) \) by first checking if \( y \) is of the form \( (x, 1^{|x|^f}) \) in polytime, and then running \( M \) on \( x \) and returning the answer computed in time \( O(2^{2|y|}) \). By assumption \( \text{E} \) is closed under polynomial many-to-one reductions, so \( A \in \text{E} \). Therefore \( \text{EXP} \subseteq \text{E} \). Since \( \text{E} \subseteq \text{EXP} \), we have \( \text{E} = \text{EXP} \).

But by the time-hierarchy theorem there exists a language \( L \) such that any algorithm deciding \( L \) must run in time greater than \( 2^{n^2} \) for infinitely many inputs, but \( L \) can be decided by an algorithm in \( \text{DTIME}(2^{O(n^2)}) \) so \( L \in \text{EXP} \). Thus, \( L \in \text{EXP} \) but \( L \notin \text{E} \) so \( \text{E} \subsetneq \text{EXP} \). This contradicts the conclusion that \( \text{E} = \text{EXP} \) so we conclude our assumption that \( \text{E} \) was closed under polynomial many-to-one reductions was false.

Finally, since \( \text{E} \) is not closed under polynomial many-to-one reductions and \( \text{NP} \) is closed under polynomial many-to-one reductions, it immediately follows that \( \text{NP} \neq \text{E} \).

**Problem 3**

Show that \( \text{SPACE}(n) \neq \text{NP} \).

**Solution to Problem 3**

Assume that \( \text{SPACE}(n) = \text{NP} \). Then \( \text{SPACE}(n) \) must be closed under polynomial many-to-one reductions since \( \text{NP} \) is closed under polynomial many-to-one reductions as shown below. Consider an arbitrary language \( A \in \text{SPACE}(n^2) \). As shown above, \( A \leq_m A_{n^2} \). Furthermore, \( A_{n^2} \in \text{SPACE}(n) \) since we can decide \( y = (x, 1^{|x|^2}) \in A_{n^2} \) by running the algorithm for deciding \( A \) on \( x \), and returning the answer computed. This will use \( O(n) \) space since \( |x|^2 = \text{O}(|y|) = \text{O}(n) \). Then \( A \in \text{SPACE}(n) \) since \( \text{SPACE}(n) \) is closed under polynomial many-to-one reductions by assumption. Thus, \( \text{SPACE}(n^2) \subseteq \text{SPACE}(n) \) so \( \text{SPACE}(n) = \text{SPACE}(n^2) \) since the other direction of containment trivially holds.
By the space-hierarchy theorem $\text{SPACE}(n) \subsetneq \text{SPACE}(n^2)$. This is a contradiction so our assumption that $\text{SPACE}(n) = \text{NP}$ must be false, and we conclude that $\text{SPACE}(n) \neq \text{NP}$.

**Problem 4**

Prove that if $\text{P} = \text{NP}$, then $\text{EXP} = \text{NEXP}$.

**Solution to Problem 4**

Assume that $\text{P} = \text{NP}$. Consider an arbitrary language $L \in \text{NEXP}$. Then $L$ is decided by a NDTM $M$ running in time $t(n) \in O(2^{cn})$ for some $c \in \mathbb{N}$. We now observe that $L_{t(n)} \in \text{NP}$ since we can decide $L_{t(n)}$ by verifying that the input given is of the form $(x, 1^{t(|x|)})$ and then running $M$ on $x$. But then by assumption, $L_{t(n)} \in \text{P}$. Let $N$ be a DTM deciding $L_{t(n)}$ in time $q(n) \in \text{poly}(n)$. We can now decide $L$ in deterministic exponential time as follows: First, replace $x$ with pad$_{t(n)}(x)$ in $\text{poly}(t(|x|))$ time. Then run $N$ on pad$_{t(n)}(x)$ in $q(O(t(|x|))) \in \text{poly}(2^{ct})$ time, accepting if and only if $N$ accepts. This algorithm runs in $O(2^{cn})$ time for some $d \in \mathbb{N}$ and decides $L$, so we conclude that $L \in \text{EXP}$. Thus, $\text{NEXP} \subseteq \text{EXP}$ and by definition we have $\text{EXP} \subseteq \text{NEXP}$, so we conclude that $\text{EXP} = \text{NEXP}$.

**Problem 5**

Suppose $L_1, L_2 \in \text{NP} \cap \text{coNP}$. Show that the language

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

is in $\text{NP} \cap \text{coNP}$.

**Solution to Problem 5**

Let $L = A \triangle B$ for some $A, B \in \text{NP} \cap \text{coNP}$. We will show that $L \in \text{NP} \cap \text{coNP}$. Let $M_1$ and $M_0$ be polynomial-time NDTMs deciding $A$ and $\overline{A}$ respectively, and let $N_1$ and $N_0$ be NDTMs deciding $B$ and $\overline{B}$ respectively.

To show that $L \in \text{NP}$ we exhibit an $\text{NP}$ algorithm to decide $L$. Non-deterministically guess a bit $i$, and then simulate $M_i(x)$ and $N_{1-i}(x)$, and accept if both $M$ and $N$ both accept. Now $x \in L$ if and only if either $x \in A \cap \overline{B}$ or $x \in \overline{A} \cap B$ if and only if there is a choice of $i$ such that $M_i(x)$ and $N_{1-i}(x)$ both accept.

To show that $L \in \text{coNP}$, we’ll show that $\overline{L} \in \text{NP}$ by exhibiting an almost identical algorithm, modified to simulate $M_i(x)$ and $N_i(x)$ for a given choice of $i$. Correctness follows by an analogous argument.