CPSC 511/611 — Solutions for Assignment #4

Solutions

1. In this question, you were asked to suppose that # s a symbol that does not belong to an alphabet Σ; let $\widehat{\Sigma} = \Sigma \cup \{\#\}$. Let

   $$\textit{pad}\colon \Sigma^\star \times \mathbb{N} \to \widehat{\Sigma}^\star$$

   be a function such that, for every string $\omega \in \Sigma^\star$ and every natural number $n \geq 0$,

   - if n is greater than or equal to the length of ω then *pad*(ω, n) is the string in $\widehat{\Sigma}^\star$ that begins with the first input string ω and that continues (and ends) with $n - |\omega|$ copies of the symbol # — so that *pad*(ω, n) has length n;
   - on the other hand, if n is less than the length of ω then *pad*(ω, n) is the first input ω, so that the length of *pad*(ω, n) is the same as the length of ω.

   Thus *pad*(ω, n) always begins with the string ω and always has length $\max(n, |\omega|)$.

   (a) Let k be an integer such that $k \geq 1$ and let Σ be any alphabet. You were asked to prove the following: If $A \subseteq \Sigma^\star$ then $A \in$ SPACE(n^k) if and only if

   $$\{\textit{pad}(\omega, |\omega|^k) \mid \omega \in A\} \in \text{SPACE}(n).$$

   Solution: Suppose, first that $A \subseteq \Sigma^\star$ such that $A \in$ SPACE(n^k). Let M be a deterministic Turing machine deciding A using work space in $O(n^k)$ in the worst case when executed on an input string in $\Sigma^\star$ with length n.
   Consider another deterministic Turing machine $\widehat{M}$ that implements the following.

   On input $\mu \in \widehat{\Sigma}^\star$:

   1. Sweeping over μ from left right, compute the unpadded decimal representation of the length n of the longest prefix ω of μ that belongs to $\Sigma^\star$.
   2. Compute the unpadded decimal representation of $n^k - n$.

3. Continue to sweep from left to right over the rest of μ, examining the symbols that are seen and computing the length of the rest of this string. If μ continues with $n^k - n$ copies of # — and nothing else — then go to step 4. Otherwise ***reject***.
4. Make a copy of ω (without the trailing #'s) on another tape, moving the tape head back of the leftmost cell of this tape.
5. Simulate the execution of M on input ω, using this new copy of ω as the input. If M ***accepts*** ω then ***accept***. If M ***rejects*** ω then ***reject***.

It should be clear, from the above description of the algorithm, that, since M decides A, $\widehat{M}$ decides the language $\{\mathit{pad}(\omega, |\omega|^k) \mid \omega \in A\}$. Now, since the length of the unpadded decimal representation of an integer n is logarithmic in n, k is a positive integer constant, and integer addition and multiplication can both be carried out using space that is linear in the length of the decimal representations of the integers to be added or multiplied, steps 1–3 can be carried out using work space that is logarithmic in the length of the input string — so that it can certainly carried out using at most linear space.

Step 4 can be carried out using space that is at most linear in the length of ω, and this is certainly at most linear in the length of the original input string μ.

Finally, the simulation of M at step 5 uses space linear in n^k, that is, linear in the length of the original input string μ as well.

Thus $\widehat{M}$ decides the language $\{\mathit{pad}(\omega, |\omega|^k) \mid \omega \in A\}$ using space at most linear in the length of the input string, as needed to establish that this language is in SPACE(n).

Now suppose, conversely, that the language $\{\mathit{pad}(\omega, |\omega|^k) \mid \omega \in A\}$ is in SPACE(n). Let $\widehat{M}$ be a deterministic Turing machine that decides this language using space that is linear in the length of the input string in the worst case. Consider a deterministic Turing machine that implements the following.

On input $\omega \in \Sigma^\star$:

1. Sweeping over ω from left to right, compute the unpadded decimal representation of the length n of this string.
2. Compute the unpadded decimal representation of $n^k - n$.
3. Write ω, followed by another $n^k - n$ copies of #, onto another tape, and move the tape head back to the leftmost cell of this tape.
4. Simulate the execution of $\widehat{M}$ on the input string that has been written onto the tape in step 3. If $\widehat{M}$ ***accepts*** then ***accepts***. If $\widehat{M}$ ***rejects*** then ***reject***.

It should be clear that M decides the language A.

Once again, steps 1 and 2 can be carried out using space that is logarithmic in the length n of ω, so that they can certainly be carried out using space that is at most linear in n^k. Since the length of the string written during step 3 has length n^k this step can be carried out using space at most linear in n^k too.

Finally, since $\widehat{M}$ uses length at most linear in the length of its input string, step 4 can be carried out using space at most linear in $n^k = |\omega|^k$ as well — as needed to prove that $A \in \mathsf{SPACE}(n^k)$.

Thus $A \in \mathsf{SPACE}(n^k)$ if and only if $\{\mathit{pad}(\omega, |\omega|^k) \mid \omega \in A\} \in \mathsf{SPACE}(n)$, as claimed.

(b) You were next asked to prove that $\mathcal{P} \neq \mathsf{SPACE}(n)$.

Solution: Suppose (in order to obtain a contradiction) that $\mathcal{P} = \mathsf{SPACE}(n)$.

Since the functions $f(n) = n$ and $g(n) = n^2$ are both space-constructible functions such that $f \in o(g)$, it follows by the Space Hierarchy Theorem that there exists a language $A \subseteq \Sigma^\star$ (defined over some alphabet Σ) such that $A \in \mathsf{SPACE}(n^2)$ but $A \notin \mathsf{SPACE}(n)$.

Now consider the language $\widehat{A} = \{\mathit{pad}(\omega, |\omega|^2) \mid \omega \in A\}$. Since $A \in \mathsf{SPACE}(n^2)$ it follows by the result of part (a) that $\widehat{A} \in \mathsf{SPACE}(n) = \mathcal{P}$.

However, it now follows that $A \in \mathcal{P}$ as well — because the second algorithm described in the solution for part (a) is a deterministic polynomial-time algorithm if $\widehat{M}$ is a deterministic Turing machine deciding the language $\widehat{A}$ using a polynomial number of steps in the worst case.

Consequently $A \in \mathcal{P} = \mathsf{SPACE}(n)$ — contradicting the choice of A as a language that is not in the latter complexity class.

The assumption made at the beginning of the proof must, therefore, be incorrect: $\mathcal{P} \neq \mathsf{SPACE}(n)$.

2. Let $L \subseteq \Sigma^\star$ be an $\mathcal{NP}$-complete language.

(a) Let

$$\widehat{\Sigma} = \Sigma \cup \{\texttt{(}, \texttt{)}, \texttt{,}, \texttt{T}, \texttt{F}\}.$$

and let

$$\widehat{L} = \{\texttt{(}\omega\texttt{,T)} \mid w \in L\} \cup \{\texttt{(}\omega\texttt{,F)} \mid \omega \in \Sigma^\star \text{ and } \omega \notin L\}.$$

You were asked to give a ***short*** proof that if $\mathcal{NP} \neq$ co-$\mathcal{NP}$ than

$$\widehat{L} \notin \mathcal{NP} \cup \text{co-}\mathcal{NP}.$$

Solution: Suppose, to obtain a contradiction, that $\mathcal{NP} \neq$ co-$\mathcal{NP}$ but that $\widehat{L} \in \mathcal{NP} \cup$ co-$\mathcal{NP}$. Then either $\widehat{L} \in \mathcal{NP}$ or $\widehat{L} \in$ co-$\mathcal{NP}$. These cases are considered separately below.

Suppose first that $\widehat{L} \in \mathcal{NP}$. Then the complement $\overline{L}$ of L is in $\mathcal{NP}$ as well, since the function $f : \Sigma^\star \to \widehat{\Sigma}^\star$ such that

$$f(\omega) = \texttt{(}\omega\texttt{,F)}$$

is clearly a polynomial-time mapping reduction from $\overline{L}$ to $\widehat{L}$, and $\mathcal{NP}$ is closed with respect to these reductions.

Since $\overline{L}$ is co-$\mathcal{NP}$-complete, L' is polynomial-time mapping reducible to $\overline{L}$ for every language $L' \in$ co-$\mathcal{NP}$. It follows that $L' \in \mathcal{NP}$ (since $\overline{L} \in \mathcal{NP}$, as above) for every such language, so that co-$\mathcal{NP} \subseteq \mathcal{NP}$.

However, a consideration of the complements of languages confirms that it must now be true that $\mathcal{NP} \subseteq$ co-$\mathcal{NP}$ as well, so that $\mathcal{NP} =$ co-$\mathcal{NP}$ — ***contradicting*** the assumption in the claim. This case is, therefore, impossible.

Suppose next that $\widehat{L} \in$ co-$\mathcal{NP}$. Then $L \in$ co-$\mathcal{NP}$ as well, since the function $f : \Sigma^\star \to \widehat{\Sigma}^\star$ such that

$$f(\omega) = (\omega\,\texttt{,T})$$

is clearly a polynomial-time mapping reduction from L to $\widehat{L}$, and co-$\mathcal{NP}$ is closed with respect to these reductions.

Since L is $\mathcal{NP}$-complete, L' is polynomial-time mapping reducible to L for every language $L' \in \mathcal{NP}$. It follows that $L' \in$ co-$\mathcal{NP}$ (since $L \in$ co-$\mathcal{NP}$, as noted above) for every such language, so that $\mathcal{NP} \subseteq$ co-$\mathcal{NP}$.

However, a consideration of the complements of languages confirms that it must now be true that co-$\mathcal{NP} \subseteq \mathcal{NP}$ as well, so that $\mathcal{NP} =$ co-$\mathcal{NP}$ — ***contradicting*** the assumption in the claim. This case is, therefore, impossible as well.

Since a contradiction has been obtained in every possible case, this establishes the claim.

(b) You were then asked to use the above to argue that if $\mathcal{NP} \neq$ co-$\mathcal{NP}$ then

$$\mathcal{NP} \cup \text{co-}\mathcal{NP} \subsetneq \mathcal{P}^{SAT}.$$

Solution: We will show, first, that $\mathcal{NP} \subseteq \mathcal{P}^{SAT}$. To see that this is the case, let $L \subseteq \Sigma^\star$ be a language such that $L \in \mathcal{NP}$. Now, since SAT is $\mathcal{NP}$-complete, there exists a polynomial-time mapping reduction $f : \Sigma^\star \to \Sigma_{\text{SAT}}^\star$ (where SAT $\subseteq \Sigma_{\text{SAT}}^\star$) from L to SAT. It should now be reasonably clear that the following algorithm can be implemented using a deterministic polynomial-time Turing machine, with an oracle for SAT, that decides L:

```
On input $\omega \in \Sigma^\star$:
1.  Compute the string $f(\omega) \in \Sigma_{\text{SAT}}^\star$
2.  if ($f(\omega) \in$ SAT) {
3.    accept
    } else {
4.    reject
    }
```

Thus $L \in \mathcal{P}^{\text{SAT}}$. Since L was arbitrarily chosen from $\mathcal{NP}$ it follows that $\mathcal{NP} \subseteq \mathcal{P}^{\text{SAT}}$, as claimed.

Next we will show that co-$\mathcal{NP} \subseteq \mathcal{P}^{\textit{SAT}}$ as well. To see that this is the case, let $L \subseteq \Sigma^\star$ be a language such that $L \in$ co-$\mathcal{NP}$. Then the complement $\overline{L}$ of L is in $\mathcal{NP}$. Now, since SAT is $\mathcal{NP}$-complete, there exists a polynomial-time mapping reduction $f : \Sigma^\star \to \Sigma^\star_{\text{SAT}}$ from $\overline{L}$ to SAT. This is also a polynomial-time mapping reduction from L to $\overline{\text{SAT}}$. It should now be reasonably clear that the following algorithm can be implemented using a deterministic polynomial-time Turing machine, with an oracle for SAT, that decides L.

```
On input $\omega \in \Sigma^\star$:
1.  Compute the string $f(\omega) \in \Sigma^\star_{\text{SAT}}$
2.  if ($f(\omega) \in$ SAT) {
3.    reject
    } else {
4.    accept
    }
```

Thus $L \in \mathcal{P}^{\text{SAT}}$. Since L was arbitrarily chosen from co-$\mathcal{NP}$ it follows that co-$\mathcal{NP} \subseteq \mathcal{P}^{\text{SAT}}$, as claimed.

It therefore follows that

$$\mathcal{NP} \cup \text{co-}\mathcal{NP} \subseteq \mathcal{P}^{\text{SAT}}.$$

Now consider the language $\widehat{L} \subseteq \widehat{\Sigma}^\star$ defined in part (a) of this question, and let $L \subseteq \Sigma^\star$ be the $\mathcal{NP}$-complete language that is used to define it. Since $L \in \mathcal{NP}$ there exists a polynomial-time mapping reduction $f : \Sigma^\star \to \Sigma^\star_{\text{SAT}}$ from L to SAT. It should now be reasonably clear, from the definition of $\widehat{L}$, that the following algorithm can be implemented using a polynomial-time deterministic Turing machine with an oracle for SAT that decides the language $\widehat{L}$:

```
On input $\mu \in \widehat{\Sigma}^\star$:
1.  if ($\mu$ is ($\omega$,T) for some string $\omega \in \Sigma^\star$) {
2.    Compute the string $f(\omega) \in \Sigma^\star_{\text{SAT}}$
3.    if ($f(\omega) \in$ SAT) {
4.      accept
      } else {
5.      reject
      }
6.  } else if ($\mu$ is ($\omega$,F) for some string $\omega \in \Sigma^\star$) {
7.      Compute the string $f(\omega) \in \Sigma^\star_{\mathsf{SAT}}$
8.      if ($f(\omega) \in$ SAT) {
9.          reject
        } else {
10.         accept
        }
    } else {
11      reject
    }
```

It follows that $\widehat{L} \in \mathcal{P}^{\mathsf{SAT}}$. Since $\widehat{L} \notin \mathcal{NP} \cup \text{co-}\mathcal{NP}$ if $\mathcal{NP} \neq \text{co-}\mathcal{NP}$, this implies that

$$\mathcal{NP} \cup \text{co-}\mathcal{NP} \subsetneq \mathcal{P}^{\mathsf{SAT}}$$

if $\mathcal{NP} \neq \text{co-}\mathcal{NP}$, as claimed.