Lecture #10: Nondeterministic Time — More about $\mathcal{NP}$, and co-$\mathcal{NP}$

Lecture Presentation

The following result was stated as “Claim #4(a)” in the notes for this lecture.

Claim. *The complexity class $\mathcal{NP}$ is closed under polynomial-time many-one reductions.*

Proof of This Claim:

Consider, once again, the language $A_{\mathsf{NTM+I+Time}} \subseteq \Sigma^{\star}_{\mathsf{UTM}}$ of encodings of nondeterministic multi-tape Turing machines

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathsf{accept}}, q_{\mathsf{reject}}),$$

input strings $\omega \in \Sigma^{\star}$, and positive integers T (encoded in unary) such that M accepts ω using at most T steps. It is proved, in the lecture notes, that this language is $\mathcal{NP}$-complete.

Consider, as well, the language $NA_{\mathsf{NTM+I+Time}} \subseteq \Sigma^{\star}_{\mathsf{UTM}}$ of encodings of nondeterministic multi-tape Turing machines

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\mathsf{accept}}, q_{\mathsf{reject}}),$$

input strings $\omega \in \Sigma^{\star}$. and positive integers T (encoded in unary) such that M *does not* accept ω using at most T steps — that is, there is no accepting configuration in the computation tree for ω (and the nondeterministic Turing machine) whose distance from the root is at most T.

How are These Languages Related? Is One the Complement of the Other?

Claim. *The language $NA_{\text{NTM+I+Time}}$ is co-$\mathcal{NP}$-complete.*

Proof:

Finally, let $L \subseteq \Sigma^\star$, where Σ is an alphabet such that the symbol "♠" does not belong to Σ — and suppose that $L \in \mathcal{P}$ — but $L \neq \emptyset$ and $L \neq \Sigma^\star$, so that there exist strings $\omega_{\text{no}}, \omega_{\text{yes}} \in \Sigma^\star$ such that $\omega_{\text{no}} \notin L$ and $\omega_{\text{yes}} \in L$.

Now consider the alphabet $\widetilde{\Sigma} = \Sigma \cup \{\spadesuit\}$, and let

$$\widetilde{L} = \{\mu \in \widetilde{\Sigma}^\star \mid \mu = \omega_\ell \spadesuit^T \text{ for a string } \omega_\ell \in \Sigma^\star \text{ and a positive integer } T \text{, for which}$$
$$\text{there exists a string } \omega_r \in \Sigma^\star \text{, such that } |\omega_r| \leq T \text{ and } \omega_\ell \cdot \omega_r \in L\}.$$

Prove that $\widetilde{L} \in \mathcal{NP}$.

Now let $\Sigma = \Sigma_{\mathsf{UTM}} \cup \{\,\widehat{\#}\,\}$[1] and suppose that $L \subseteq \Sigma^\star$ is the language of strings

$$\mu\,\widehat{\#}\,\nu$$

where $\mu \in A_{\mathsf{NTM+I+Time}}$, and $\nu \in \Sigma^\star_{\mathsf{UTM}}$ is a short certificate for μ as described in the lecture notes (so that $|\nu| \le |\mu|^2$). Then $L \in \mathcal{P}$, since L is the language of the polynomial-time verification algorithm for $A_{\mathsf{NTM+I+Time}}$ that is described in the lecture notes.

Prove that — if L is the above language and $\widetilde{L} \subseteq \widetilde{\Sigma}^\star$, where $\widetilde{\Sigma} = \Sigma \cup \{\spadesuit\} = \Sigma_{\mathsf{UTM}} \cup \{\,\widehat{\#}, \spadesuit\}$ and $\widetilde{L}$ is the language obtained from L by the construction that has now been described — then $\widetilde{L}$ is $\mathcal{NP}$-complete.

OK, OK, OK... The lecture notes suggest that we only know ***one*** $\mathcal{NP}$-complete language, and the completion of this exercise supplies a second one. (This exercise was added *after* the lecture notes had been completed.)

This does not really change (or undermine) the point now being made: The $\mathcal{NP}$-complete languages we know about are highly artificial, and ask questions that are about computations by nondeterministic Turing machines. They do not provide very much evidence that *other*, more natural, $\mathcal{NP}$-complete languages will be found (so that this "theory" will be of any practical use).

[1]Since the symbol "#" already belongs to Σ_{UTM}, the separator, used in the definition of a polynomial-time verification algorithm, is "$\widehat{\#}$", so that it is a new symbol.