Exercise Sheet 4: Time Complexity

David Carral

November 27, 2019
Exercise 1.1

Show that $P$ is closed under concatenation.

**Solution.** Consider some languages $A, B \in P$. We show that the union of these two languages is in $P$. That is, we show that $L = \{uv \mid u \in A \text{ and } v \in B\}$ is in $P$.

1. Since $A, B \in P$, there are some poly-time bounded TMs $M$ and $N$ that solve $A$ and $B$, respectively.
2. Let $D$ be the 3-tape TM that performs the following computation on input $w$:
   - Sequentially iterate through each $i \in \{0, \ldots, |w|\}$:
     - Copy $w_0,i$ on to working tape 1 and simulate $M$ on this input.
     - Copy $w_i,|w|$ on to working tape 2 and simulate $N$ on this input.
     - If both simulations accept, then accept.
     - Clear both working tapes.
   - Reject.
3. $D$ accepts $L$ (discuss).
4. $D$ is a poly-time bounded TM (discuss).
Exercise 1.2

Show that $P$ is closed under star.

Solution. Given some language $L \in P$, we show that $L^* \in P$.

1. There is a poly-time bounded TM $M$ that solves $L$.
2. Let $D$ be the 3-tape DTM that performs the following computation on input $w$:
   - If $w$ is the empty word, then accept.
   - For every $0 \leq i < j \leq |w|$
     - Copy the string $w_{i,j}$ on to working tape 1.
     - Simulate $M$ on working tape 1. If this simulation accepts, append the pair $(i, j)$ on to working tape 2.
     - Clear tape 1.
   - Accept if and only if there is a path from 0 to $|w|$ on the directed graph represented in working tape 2.
3. $D$ accepts $L^*$ (discuss).
4. $D$ is a poly-time bounded TM (discuss).
Consider the problem \textit{CLIQUE}:

- Input: An undirected graph $G$ and some $k \in \mathbb{N}$
- Output: Does there exist a clique in $G$ of size at least $k$?

Suppose \textit{CLIQUE} can be solved in time $T(n)$ for some $T: \mathbb{N} \rightarrow \mathbb{N}$ with $T(n) \geq n$ for all $n \in \mathbb{N}$. Then, show that the optimisation problem

- Input: An undirected graph $G$.
- Output: A clique in $G$ of maximal size.

can be computed in time $O(n \cdot T(n))$. You can assume that $T$ is monotone.
Exercise 2

Suppose \textit{CLIQUE} can be solved in time $T(n)$ for some $T : \mathbb{N} \rightarrow \mathbb{N}$ with $T(n) \geq n$ for all $n \in \mathbb{N}$. Then, show that the optimisation problem

\begin{itemize}
  \item Input: An undirected graph $G$.
  \item Output: A clique in $G$ of maximal size.
\end{itemize}

\textit{can be computed in time $O(n \cdot T(n))$.} You can assume that $T$ is monotone.

\textbf{Solution.} For an input graph $G = \{V, E\}$, apply the following strategy.

\begin{itemize}
  \item Using binary search and the $T(n)$-procedure to solve \textit{CLIQUE}, compute the size $k$ of a maximal clique in $G$.
  \item Sequentially iterate through every $v \in V$ and perform the following computation:
    \begin{itemize}
      \item Compute the graph $G'$ that results from removing the vertex $v$ from $G$.
      \item Check if $G'$ contains a clique of size $k$.
      \item If so, then remove the vertex $v$ from $G$.
    \end{itemize}
  \item Return $G$.
\end{itemize}

\textbf{Remark:} The above strategy runs the decision procedure of \textit{CLIQUE} at most $\lceil \log n \rceil$ times to compute $k$ plus at most $n$ times to weed out nodes.
Exercise 3

Show that if a language $L$ is $\text{NP}$-complete, then $\overline{L}$ is $\text{coNP}$-complete.

Solution.

**A.** If $L$ is in $\text{NP}$, then $\overline{L}$ is in $\text{coNP}$.

**B.** We show that, if $L$ is $\text{NP}$-hard, then $\overline{L}$ is $\text{coNP}$-hard.

1. $L$ is $\text{NP}$-hard.
2. Let $K$ be a language in $\text{coNP}$.
3. By (2): $\overline{K}$ is in $\text{NP}$.
4. By (1) and (3): $\overline{K} \leq_p L$.
5. By (4): $K \leq_p \overline{L}$.
6. By (2) and (5): $\overline{L}$ is $\text{coNP}$-hard.

**C.** By (A) and (B): if $L$ is $\text{NP}$-complete, then $\overline{L}$ is $\text{coNP}$-complete.
Exercise 4

Show that if $P = NP$, then a polynomial-time algorithm exists that produces a satisfying assignment of a given satisfiable propositional formula.

Solution. Step by step solution.

1. Assume that $P = NP$.
2. $Sat \in NP$.
3. By (1) and (2): $Sat \in P$.
4. Let $\phi$ be some satisfiable propositional formula with variables $X_1, \ldots, X_n$.
5. For a sequence $V = V_1, \ldots, V_m$ with $m \leq n$ and $V_i \in \{0, 1\}$ for all $1 \leq i \leq m$, let $\phi_V$ be the formula that results from replacing $X_i$ by $V_i$ in $\phi$ for all $1 \leq i \leq m$.
6. Let $U_1 = 0$, if $\phi_0$ is satisfiable; and $U_1 = 1$, otherwise.
7. For all $2 \leq i \leq n$, let $U_i = 0$ if $\phi_{U_1, \ldots, U_{i-1}, 0}$ is satisfiable, and $U_i = 1$ otherwise.
8. We can show via induction that, for all $1 \leq i \leq n$, the formula $\phi_{U_1, \ldots, U_i}$ for all $1 \leq i \leq n$ is satisfiable.
9. The assignment $\alpha$ mapping $X_i$ to $U_i$ for all $1 \leq i \leq n$ is satisfying for $\phi$.
10. The assignment $\alpha$ can be computed in polynomial time (discuss).
Show that the following problem is NP-complete.

\[ \text{Path} = \{ \langle G, s, t, k \rangle \mid G \text{ contains a simple path from } s \text{ to } t \text{ of length } k \} \]

Solution. Let us have a look at a similar problem.

\[ \text{AtMostPath} = \{ \langle G, s, t, k \rangle \mid G \text{ contains a simple path from } s \text{ to } t \text{ of length at most } k \} \]

Is \text{AtMostPath} in NP? In P? Discuss.

Remark. If there is a (possibly non-simple) path in \(G\) of length at most \(k\) from \(s\) to \(t\), then there is a simple path in \(G\) of length at most \(k\) from \(s\) to \(t\).
Exercise 5

Show that the following problem is \( \text{NP} \)-complete.

\[
Path = \{ \langle G, s, t, k \rangle \mid G \text{ contains a simple path from } s \text{ to } t \text{ of length } k \}
\]

**Solution.** The above problem is in \( \text{NP} \) (discuss).

**Definition.** \( G \in \text{HamCycle} \) iff there is a path in \( G \) that starts and ends in the same vertex and visits each vertex in \( G \) exactly once.

To show that it is also \( \text{NP} \)-hard, we show that \( \text{HamCycle} \leq_p \text{Path} \).

1. Let \( f \) be the function mapping \( \langle G \rangle \) into \( \langle G, s, s, n \rangle \) with \( s \) the first vertex syntactically occurring in \( \langle G \rangle \) and \( n \) is the number of vertices in \( G \).
2. By (1): the function \( f \) is poly-time computable.
3. By (1): if \( \langle G \rangle \in \text{HamCycle} \), then \( \langle G, s, s, n \rangle \in \text{Path} \) (discuss).
4. By (1): if \( \langle G, s, s, n \rangle \in \text{Path} \), then \( \langle G \rangle \in \text{HamCycle} \) (discuss).
5. By (2), (3), and (4): \( \text{HamCycle} \leq_p \text{Path} \).
Exercise 6

Let $A \subseteq 1^*$. Show that, if $A$ is NP-complete, then $P = NP$.

**Solution** To show that $P = NP$, we describe a polynomial-time procedure for $SAT$.

- Let $f$ be a poly-time reduction from $SAT$ to $A$.
- Let $\phi$ be a propositional formula and $\{x_1, \ldots, x_n\}$ be the set of all variables in $\phi$.
- We inductively define the sets $L_0, \ldots, L_n$:
  1. $L_0 = \{\langle f(\phi), \phi \rangle\}$
  2. For every $k \in \{1, \ldots, n\}$ and $\langle f(\psi), \psi \rangle \in L_{k-1}$:
     - Add $\langle f(\psi_{x_k=0}), \psi_{x_k=0} \rangle$ and $\langle f(\psi_{x_k=1}), \psi_{x_k=1} \rangle$ to $L_k$ where $\psi_{x_k=\ell}$ denotes the formula $\psi$ with the variable $x_k$ set to the value $\ell$ and simplified afterwards.
     - Then, remove all pairs $\langle w, \psi \rangle$ in $L_k$ where $w$ contains some symbol different from $1$.
     - Furthermore, for every $m \in \mathbb{N}$, keep at most one (arbitrarily chosen) pair of the form $\langle 1^m, \psi \rangle$ in $L_k$.

- Then, $\phi$ is satisfiable if and only if $L_n \neq \emptyset$.

Can we compute the sets $L_0, \ldots, L_n$ in polynomial time in the size of $\phi$? Discuss