Computability and Computational Complexity, A.Y. 2023–2024

Guide to the answers

Monday, June 17, 2024

Exercise 1

Remember that by "vertex coloring" of an undirected graph we mean the task of assigning an element from a finite set of labels ("colors") to each vertex of the graph so that no two connected vertices have the same color; we say that a graph can be k-colored if the task can be successfully carried out with no more than k colors.
For each of the following coloring-related languages, determine whether they are in **P**, **NP** and/or **coNP** and provide a short motivation for your answers:

1. graphs that can be 2-colored;
2. graphs that cannot be 2-colored;
3. graphs that can be 3-colored;
4. graphs that cannot be 3-colored;
5. graphs $G = (V, E)$ that cannot be k-colored for any $k < |V|$.

Acceptable motivations can be formulated as short descriptions of algorithms (verbal or any kind of pseudocode), set inclusions (e.g., "I already proved that $L \in A$, but we know that $A \subset B$, therefore $L \in B$"), reductions to some **NP**-complete version of SATISFIABILITY (e.g., 3-SAT).

Solution 1

1. A simple greedy algorithm is sufficient to check whether a graph can be 2-colored, since the only arbitrary choice is the color of the starting node (for each connected component); therefore, the language is in **P**. As **P** $\subseteq$ **NP** and **P** $\subseteq$ **coNP**, the language is also in **NP** and in **coNP**.
2. Same as above: if a language is in **P**, its complement is in **P** too.
3. 3-VERTEX-COLORING is clearly **NP**, since a legal 3-coloring of a graph can be checked in polynomial time.
   Moreover, it is a well known **NP**-complete language, see the notes for details of the reduction from 3SAT. As such, it is not not known to be in **P**. Likewise, no polynominomial certificate is known for a graph *not* being 3-colorable, therefore the language in not known to be in **coNP**.
4. This is the complement of the previous language, therefore it belongs to **coNP**, but for the same reasons as above it is not known to belong to **P** nor **NP**.
5. The only graphs that *need* $k = |V|$ colors are the complete graphs, for which we must use a different color for each vertex.
   In all other cases, it is always possible to use $k = |V| - 1$ colors by assigning the same color to two nodes not connected by an edge, and different colors to all others.

Therefore the proposed language coincides with the set of complete graphs, and a graph's completeness can be checked in polynomial time.
Hence the language lies in **P**, **NP** and **coNP**.

$\boxed{\textbf{Exercise 2}}$

Choose **one** of the following topics and develop it along the suggested lines.

Topic 1: **Kolmogorov complexity**

1. Given a suitable alphabet Σ and a universal Turing machine (UTM) $\mathcal{U}$, define the Kolmogorov complexity $K_{\mathcal{U}}(s)$ of a string $s \in \Sigma^*$.

2. Show that, given two UTMs $\mathcal{U}$ and $\mathcal{V}$, the difference between the Kolmogorov complexities $K_{\mathcal{U}}(s)$ and $K_{\mathcal{V}}(s)$ of the same string s is bounded by a constant that only depends on $\mathcal{U}$ and $\mathcal{V}$ and not on s.

3. Outline the proof that $K_{\mathcal{U}}$ is uncomputable.

Topic 2: **Post Correspondence Problem**

1. Define the Post Correspondence Problem (PCP).

2. Given the "Restricted" version of the Post Correspondence Problem (RPCP) where the initial pair is fixed (i.e., pre-set as part of the instance), show how an instance of RPCP can be reduced to an equivalent instance of PCP.

3. Outline the proof that PCP is uncomputable.

$\boxed{\textbf{Solution 2}}$

See the lecture notes.