CPSC 511 — Winter, 2024
Assignment #2 — $\mathcal{NP}$-Completeness

About This Assignment

This assignment is due by 11:59 pm on Friday, February 16. CPSC 511 students may either complete this assignment or individually, or working in groups of two students.

The assignment should be submitted as a single PDF file, using the D2L dropbox for this assignment. While repeated submissions are allowed, only the most recent submission will be maintained and marked.

Problems To Be Solved

Cormen, Leiserson and Stein note the following in their text, "Introduction to Algorithms": 'Map-makers try to use as few colours as possible when colouring countries on a map, as long as no two countries that share a border have the same colour." A map can be abstractly modelled as an undirected graph, where countries are represented by vertices and there is an edge between two vertices that have a common border. This motivates the ***graph colouring problem*** of finding a way to assign colours to the vertices in a graph, using as few colours as possible, in such a way that ***no pair of vertices with an edge between them receive the same colour.*** We say that an undirected graph is ***k-colourable*** if this is the case when at most k colours are used. Furthermore if a graph $G = (V, E)$ is k-colourable, then any assignment of (at most) k colours to the vertices that satisfies the above constraint is called a ***k-colouring*** of G.

With that noted, considering the following decision problem.

Graph 3-Colourability

Instance: An undirected graph $G = (V, E)$
Question: Is G 3-colourable?

This can be used to produce language of interest using the alphabet Σ_G and the encodings of undirected graphs described in Lecture #12: Let $L_{\text{Graph}} \subseteq \Sigma_G^\star$ be the set of (well-formed) encodings of undirected graphs, that is, the set of encodings of *instances* of the above problem. Let $L_{\text{3-Colurable}} \subseteq L_{\text{Graph}}$ be the set of encodings of "Yes-instances" of this problem, that is, the set of encodings of undirected graphs that are 3-colourable.

Using information from Lecture #12, one can see that $L_{\text{Graph}} \in \mathcal{P}$. This fact can be used when completing this assignment, without proving it.

1. Prove that $L_{\text{3-Colurable}} \in \mathcal{NP}$.

   It is not necessary to describe a Turing machine in order to do this: Instead, an algorithm can be given as pseudocode (when one is needed). After the correctness and efficiency of this high-level algorithm have been established, details can be added (as needed) to establish that a Turing machine, that would be needed to establish this claim, does exist.

Now consider a Boolean formula $\mathcal{F}$, including a finite number of the variables $x_0, x_1, x_2, \ldots,$ that is in 3-conjunctive normal form. Suppose, in particular, that

$$\mathcal{F} = (\ell_{1,1} \vee \ell_{1,2} \vee \ell_{1,3}) \wedge (\ell_{2,1} \vee \ell_{2,2} \vee \ell_{2,3}) \wedge \cdots \wedge (\ell_{k,1} \vee \ell_{k,2} \vee \ell_{k,3})$$

for some positive integer k.

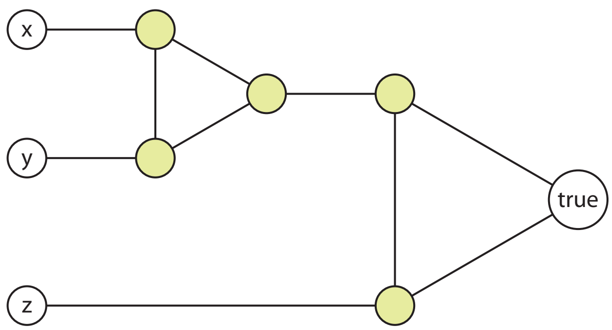
Consider an undirected graph $G_{\mathcal{F}} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ that is as follows. (For now, vertices will be given different names than $v_0, v_1, v_2, \ldots$ — but these vertices can be renamed later.)

- $V_{\mathcal{F}}$ includes three "special" vertices, "true", "false", and "neither", and $E_{\mathcal{F}}$ includes all three edges $(\text{true}, \text{false})$, $(\text{true}, \text{neither})$, and $(\text{false}, \text{neither})$ — so that these three special vertices form a triangle in the graph.
- For every Boolean variable x_h such that either x_h or $\neg x_h$ (or both) is a literal in $\mathcal{F}$, $V_{\mathcal{F}}$ also includes a pair of vertices, "x_h" and "$\neg x_h$". $E_{\mathcal{F}}$ also includes all three of the edges $(x_h, \text{neither})$, $(\neg x_h, \text{neither})$, and $(x_h, \neg x_h)$, where "neither" is the special vertex given above.

Note that, if we have a 3-colouring, then we can name the three colours anything we want to, and the three special vertices must have different colours. So, we could name the colour given to neither "`N`", we could name the colour given to true "`T`", and we could name the colour given to false "`F`".

Additional vertices and edges will be added to this graph. However, if you think about it, you should see that every 3-colouring of $G_{\mathcal{F}}$ must already correspond to some (not necessarily satisfying) truth assignment for $\mathcal{F}$.

Suppose now that we include *five* more vertices to $V_{\mathcal{F}}$ for each one of the k clauses in $\mathcal{F}$, and also add some edges to $E_{\mathcal{F}}$. In particular, Suppose that we create a subgraph looking like this for each clause:



In particular, for the i^{th} clause $(\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3})$, the vertex x shown here is the same as the vertex $\ell_{i,1}$, the vertex y is the same as the vertex $\ell_{i,2}$, and the vertex z shown here is the same as the vertex $\ell_{i,3}$ — each of which is one of the vertices "x_h" or "$\neg x_h$" that has already been added to $V_{\mathcal{F}}$ above. The vertex labelled "true" here is the special vertex "true" that has been added above, as well. The five new vertices, for this clause, are the five unnamed ones in the picture.

The set $E_{\mathcal{F}}$ should be increased (only) to include the edges in the above picture for each one of the clauses in $\mathcal{F}$. The description of the graph $G_{\mathcal{F}} = (V_{\mathcal{F}}, E_{\mathcal{F}})$ is now complete.

2. Prove that if $\mathcal{F}$ is satisfiable then $G_{\mathcal{F}}$ is 3-colourable.

3. Prove that if $G_{\mathcal{F}}$ is 3-colourable then $\mathcal{F}$ is satisfiable.

4. Prove that $L_{\text{3-Colurable}}$ is $\mathcal{NP}$-hard. Algorithms may be specified as in previous questions, with their analyses given in the same way and at the same level of detail.

If you have successfully answered the above questions then you have completed a proof that $L_{\text{3-Colurable}}$ is $\mathcal{NP}$-complete.