CSCC63 Assignment 2

Reductions, Polytime Reductions, and $\mathcal{NP}$

Due 11:59pm, July 7

Warning: *For this assignment you may work either alone or in pairs. Your electronic submission of a PDF to Crowdmark affirms that this assignment is your own work and that of your partner, and no one else's, and is also in accordance with the University of Toronto Code of Behaviour on Academic Matters, the Code of Student Conduct, and the guidelines for avoiding plagiarism in CSCC63. Note that using Google or any other online resource is considered plagiarism.*

1. (10 marks) Consider the language

$$L_1' = \{\langle G_1 = (V, \Sigma, P, S)\rangle \mid G_1 \text{ is a CFGs and there are strings } w, w' \in \Sigma^* \text{ such that } w, w', \text{ and } ww' \in L(G_1)\}.$$

Show that L_1' is undecidable.

Note that any CFG G and string x, it is possible to decide whether $x \in L(G)$ in $\mathcal{O}(|G|^2|x|^3)$ time.

2. (10 marks) Let $B = \{0^n 1^n \mid n \in \mathbb{N}\}$. Then we can write the languge L_5 from assignment 1 as

$$L_5 = \{\langle M\rangle \mid \overline{L(M)} \subseteq B\}.$$

Let $L_2' = \{\langle M\rangle \mid |\overline{L(M)} \cap \overline{B}| < \infty\}$.

Show that $L_5 \leqslant_m L_2'$.

3. (10 marks) Show that $\overline{L_5} \leqslant_m L_2'$.

4. (5 marks) Consider the language

$$\text{FACT-RANGE} = \{\langle n, a, b\rangle \mid n, a, b \in \mathbb{N}, \text{ and } \exists k \in \mathbb{N}, a \leqslant k \leqslant b \text{ and } k \text{ divides } n\}.$$

Now, consider the following program to solve FACT-RANGE:

Find-Fact= "On $\langle n, a, b\rangle$:
1. If $(0 < n < a)$ or $(b < a)$:
2. *Reject.*
3. For $i = \max(a, 1)$ to b:
4. If $n \,\%\, i == 0$:
5. *Accept.*
6. *Reject.*

Is this a polytime algorithm? Why or why not?

5. (15 marks) Let $\text{STORAGE-BOX} = \left\{ \langle U, b, k\rangle \;\middle|\; \begin{array}{l} U \text{ is a finite set of elements such that} \\ \text{each } u \in U \text{ has a size } s(u) \in \mathbb{N}, \text{ where } b, k \in \mathbb{N}, \text{ and} \\ \text{where the elements in } U \text{ can placed into } k \text{ boxes,} \\ \text{each of which has size at most } b. \end{array} \right\}$.

   (a) (5 marks) Show that STORAGE-BOX $\in \mathcal{NP}$.

   (b) (10 marks) Assuming that SUBSET-SUM is $\mathcal{NP}$-complete, show that STORAGE-BOX is also $\mathcal{NP}$-complete.

6. (15 marks) Let HALF-PATH $= \left\{ \langle G = (V, E), s, t \rangle \;\middle|\; \begin{array}{l} G \text{ is a directed graph, } s, t \in V, \\ \text{and } G \text{ has a simple path from } s \text{ to } t \\ \text{that passes through at least } |V|/2 \text{ vertices.} \end{array} \right\}$.

   (a) (5 marks) Show that HALF-PATH $\in \mathcal{NP}$.

   (b) (10 marks) Assuming that HAM-PATH is $\mathcal{NP}$-complete, show that HALF-PATH is also $\mathcal{NP}$-complete.

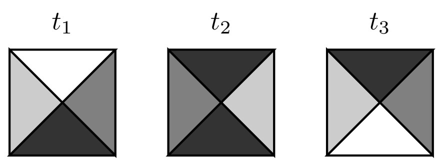
7. (15 marks) Let MONOTONE-3SAT be the language

$$\left\{ \langle \phi \rangle \;\middle|\; \begin{array}{l} \phi \text{ is a satisfiable } 3CNF \text{ Boolean formula in which} \\ \text{every clause either only contains three distinct} \\ \text{negated variables or three distinct un-negated variables.} \end{array} \right\}.$$

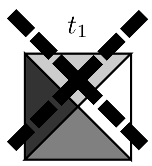
   (a) (5 marks) Show that MONOTONE-3SAT $\in \mathcal{NP}$.

   (b) (10 marks) Assuming that 3SAT is $\mathcal{NP}$-complete, show that MONOTONE-3SAT is also $\mathcal{NP}$-complete.

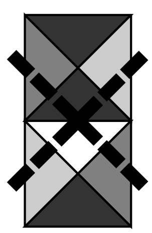
8. (20 marks) In this question we'll see an undecidable problem that is somewhat similar to the PCP. Suppose that I give you a finite set of tiles:



Our aim will be to cover the entire plain with copies of these tiles. We can re-use the tiles as many times as we want, but we cannot rotate them:

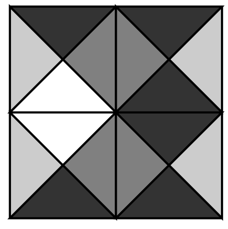


We'll have to place the copies of our tiles side-to-side, and they have to match in their colours:



Here we've tried placing t_2 dirctly above t_1 but, as you can see, the bottom colour of t_2 does not match the top of t_1.

We could, however, cover the entire plain with the tiles in our example – we'd be able to build the following 2×2 block and repeat it ad infinitum:



That is, we could use block of tiles of the form

$$\begin{matrix} t_3 & t_2 \\ t_1 & t_2 \end{matrix}$$

Note that not all tilings of the plane have to be periodic!

It turns out that it is undecidable to determine, given a set of tiles, whether it can cover the whole plane. The proof looks a bit like the PCP proof that we've gone over in class, but there are extra mechanisms we'd need which we won't go into detail about here. What we can do with the tools from class is work with a restricted version of the tiling problem.

Consider the following languge:

Definition: MULTI-TILE

> **Instance:** Finite set T of four-colour tiles of the sort seen above, along with two distinguished tiles t_a nd t_b.
>
> **Question:** Is there a way to cover the entire plain with the tiles in T in such a way that:
>
> 1) The tile t_a is used at least once, and
> 2) the tile t_b is used an infinite number of times.
>
> There must be eactly one time on every point in the plain, and the colouring and rotation rules we've described above must be followed.

Show that MULTI-TILE is neither recognizable nor co-recognizable.

9. **Bonus** (10 marks — your mark will be rounded to the nearest multiple of 2.5)

Consider the following languge:

Definition: 3COL

> **Instance:** A graph $G = (V, E)$.
>
> **Question:** Does G have a vlid 3-colouring?
> *(i.e., a map c from V to $\{$Red, Green, Blue$\}$ such that if $\{u, v\} \in E$, then $c(u) \neq c(v)$).*

Now, suppose that, instead of asking whether such a colouring c exists, we want to know *how many* such c exist. If $|V| = n$, then the number of certificates c for 3COL is somewhere between 0 and 3^n, inclusive.

Suppose you have an oracle that will tell you whether a graph G with n nodes has at least $3^{n/2}$ colourings. Give polytime algorithm using this oracle to count the eact number of 3-colourings that G has.