Homework 3, Computational Complexity 2024

The deadline is 23:59 on Wednesday 4 December. Please submit your solutions on Moodle. Typing your solutions using LaTeX is strongly encouraged. The problems are meant to be worked on in groups of 2–3 students. Please submit only one writeup per team. You are strongly encouraged to solve these problems by yourself. If you must, you may use books or online resources to help solve homework problems, but you must credit all such sources in your writeup and you must never copy material verbatim.

1 For functions $f\colon \{0,1\}^n \to \{0,1\}$ and $g\colon \{0,1\}^m \to \{0,1\}$ define their composition $f \circ g$ as the function $\{0,1\}^{nm} \to \{0,1\}$ given by

$$(f \circ g)(x) := f(g(x^1), \dots, g(x^n)) \qquad \text{where} \qquad x = (x^1, \dots, x^n) \in (\{0,1\}^m)^n.$$

(In the lecture, we discussed the special case of $f = \textsc{And}$ and $g = \textsc{Or}$.) Show that decision tree complexity D^{dt} behaves multiplicatively under composition:

$$\mathsf{D}^{\mathsf{dt}}(f \circ g) = \mathsf{D}^{\mathsf{dt}}(f) \cdot \mathsf{D}^{\mathsf{dt}}(g).$$

(*Hint: For the "$\geq$" direction, suppose we are given adversary strategies for f and g—how to design an adversary strategy for $f \circ g$?*)

2 Let us define a CNF formula F. We have a set of nodes $[n] = \{1, \dots, n\}$ and each node $i \in [n]$ is associated with $\log n$ boolean variables $x^i \in \{0,1\}^{\log n}$. We identify the set $\{0,1\}^{\log n}$ with $[n]$ so that each x^i encodes a *pointer* to another node in $[n]$. Thus there are altogether $n \log n$ many variables $x = (x^1, \dots, x^n)$. Each assignment x can be visualised as a directed graph $G_x = ([n], E)$ where $(i, j) \in E$ iff i points to j (that is, $x^i = j$). We add the following constraints to F:

1. First node points forward: $x^1 > 1$.
2. Every node points either to itself (selfloop) or forward: $x^i \geq i$ for all $i \in [n]$.
3. No node points forward to a selfloop: $(x^i > i) \to (x^{x^i} \neq x^i)$ for all $i \in [n]$.

Note that each constraint above involves only $O(\log n)$ variables and hence they can be encoded as a $O(\log n)$-width CNF of size polynomial in n. This defines the formula F.

First, show that F is unsatisfiable. Second, show that any tree-like resolution proof of F requires $\Omega(n)$ depth.

[***Update**: Turns out (thanks to Luca!) the formula F admits tree-like resolution refutations of polynomial size. (For additional fun, prove this.) Thus, F is an example showing that tree-like resolution proofs cannot be efficiently converted into* balanced trees*: there is a tree-like proof of size $S = \mathrm{poly}(n)$ but no proof of depth $\log S = O(\log n)$.*]

3 Give a reduction from the *Set-Intersection* problem (to be discussed in Lecture 11) to show that the following two-party communication problem requires $\Omega(n)$ bits of (randomised) communication:

- Alice holds a graph $G_A = ([n], E_A)$;
- Bob holds a graph $G_B = ([n], E_B)$;
- Decide whether their union $G_A \cup G_B = ([n], E_A \cup E_B)$ contains a perfect matching.