Schedule S1, (Computer Science, CS and Philosophy, Maths and CS)
Schedule B, MSc in Advanced Computer Science

Hilary Term 2022

Computational Complexity

Exercise class 1: background (TMs, (un)decidability); polynomial-time computability

1. Arrange the following functions in increasing order of order-of-growth. (Do any of them have the same order of growth?)

   (a.) n^{10} (b.) $(n^2)^{\log n}$ (c.) $(1.1)^n$ (d.) $(\log n)^{\log n}$

   (e.) $n^{\log n}$ (f.) $100n^2$ (g.) $n!/(2^n)$

2. Consider boolean functions whose inputs are propositional variables $x_1, \ldots, x_n$, which map n boolean values to a single output value.

   Prove that for any polynomial $p(n)$, there exist functions that cannot be written down using a formula of size at most $p(n)$. (A formula is built up from $x_1, \ldots, x_n$ and the usual connectives $\wedge$, $\vee$, $\neg$.)

3. Let $\mathcal{L}$ be a language and define a new language $\mathcal{L}'$ as follows. For every word $w \in \mathcal{L}$, include $w\mathtt{0}^{|w|^2}$ in $\mathcal{L}'$. That is, a word in $\mathcal{L}'$ consists of a word in $\mathcal{L}$, extended with a sequence of $\mathtt{0}$'s whose length is the square of the length of w.

   Prove that $\mathcal{L}$ is recognisable in polynomial time if and only if $\mathcal{L}'$ is recognisable in polynomial time. (You may assume that $\mathtt{0}$ is not in the alphabet of $\mathcal{L}$.)

4. A *Turing machine with two-sided unbounded tapes* is a Turing acceptor where the tapes are unbounded to both sides. Show that such machines can be simulated by our standard model of Turing machines.

   Note: You do not have to give the formal definition of the Turing machine. A precise description of what the machine does and how it simulates the original machine is sufficient.

5. Prove that if $\mathcal{L}$ is recognisable in polynomial time, then so is $\mathcal{L}^*$, where

$$\mathcal{L}^* := \{w \in \Sigma^* : w = w_1 w_2 w_3 \ldots w_k, \quad w_i \in \mathcal{L} \text{ for all } 1 \leq i \leq k\}$$

6. (a) Prove that the following problem is undecidable: Given a (standard encoding of a) Turing machine, the question is: does it run in polynomial time?

   (b) Suggest a definition of "polynomial-time Turing machine" having the property that such a machine is computationally easy to recognise, and such that every language in the class P is accepted by such a machine. Explain how your definition achieves this.

7. Suppose the decision version of the Clique problem

   CLIQUE
   Input: Graph G, $k \in \mathbb{N}$
   Problem: Does G have a clique of size $\geq k$?

can be solved in time $T(n)$ for some function $T : \mathbb{N} \to \mathbb{N}$ with $T(n) \geq n$.

Prove that the optimisation version

OPT-CLIQUE	
Input:	Graph G
Problem:	Compute a clique in G of maximum order

can be solved in time $O(n^c \cdot T(n))$, for some $c \in \mathbb{N}$.