Exercise III, Computational Complexity 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

Circuit complexity

1 In the CNF-Equivalence problem, the input consists of a pair of CNF formulas (φ, ψ) both defined over the same set of n variables $x = (x_1, \ldots, x_n)$. The goal is to decide if they are *equivalent*, that is, do they compute the same boolean function: $\varphi(x) = \psi(x)$ for all $x \in \{0,1\}^n$? Classify this problem as best as you can—is it in P, NP, or coNP? is it complete for any class?

2 Complete the proof of Circuit-SAT $\leq_p$ SAT from the lecture by finding, for each of the following logical predicates, an equivalent CNF formula.

(a) $y \leftrightarrow (x \vee z)$
(b) $y \leftrightarrow (x \wedge z)$
(c) $y \leftrightarrow \neg x$

3 The n-bit function $\textsc{Xor}_n \colon \{0,1\}^n \to \{0,1\}$ outputs 1 iff the number of 1-bits in the input is odd. Show that $\textsc{Xor}_n$ can be computed with a boolean circuit (gates $\vee$, $\wedge$, $\neg$) of size $O(n)$. Can you also make the circuit have *depth* (i.e., length of longest directed path) at most $O(\log n)$?

(Hint: Construct a circuit for $n = 2$ and then use many copies of that circuit for general n.)

4 Let φ be any DNF formula over n variables that computes $\textsc{Xor}_n$. Recall that $\varphi = T_1 \vee \cdots \vee T_m$ where each T_j is a *term*, that is, a conjunction of literals.

(a) Show that any term T_j either contains n distinct variables or is *contradictory*, meaning that it contains x_i and $\overline{x}_i$ for some variable x_i.

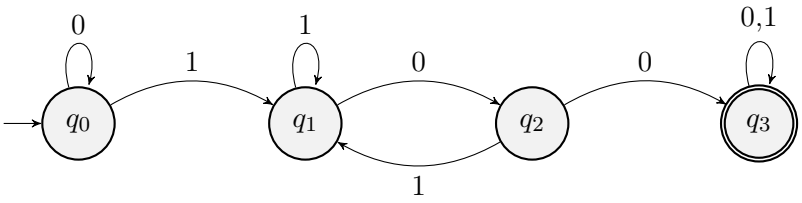
(Hint: Show that if T_j is not contradictory and omits both x_i and $\bar{x}_i$ for some i, then φ fails to compute $\textsc{Xor}_n$ correctly. Use the fact that the value of $\textsc{Xor}_n$ is flipped if we flip the value of x_i.)

(b) Show that φ must contain $m \geq 2^{n-1}$ terms.

Conclude that circuits can be exponentially more expressive than DNF/CNF formulas.

5 (*) Consider the $2n$-variate CNF formula defined by $\varphi = (x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge \ldots \wedge (x_n \vee y_n)$. Show that any DNF formula equivalent to φ requires at least 2^n terms.

6 (*) Recall from your undergrad days that a language $L \subseteq \{0,1\}^*$ is *regular* if it is accepted by a *deterministic finite automaton* $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$. Here Q is a finite set of states, $\Sigma = \{0,1\}$ is the input alphabet, $\delta\colon Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final accepting states. For example, the following automaton accepts all binary strings that contain "100" as a substring:



Prove that every regular language can be computed by a linear-size circuit. That is, let $L \subseteq \{0,1\}^*$ be a regular language. For any input length $n \in \mathbb{N}$, show how to construct a boolean circuit C_n with n input variables and $O(n)$ gates such that

$$\forall x \in \{0,1\}^n : \quad C_n(x) = 1 \iff x \in L.$$