Final Exam, Computational Complexity 2025

1 *(15 pts)* **Quick-fire round.** Consider the following statements.

1. $\mathbf{P}_{/\mathbf{poly}} \subseteq \mathbf{BPP}$.
2. $\mathbf{PostBPP} \subseteq \mathbf{PSPACE}$.
3. $\mathbf{P}^A \neq \mathbf{EXP}^A$ for every A.
4. $\mathbf{TIME}(n) \subseteq \mathbf{SPACE}(n^{0.99})$.
5. If $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{PH} = \mathbf{BPP}$.
6. If $\mathbf{PH}$ has a complete problem, then $\mathbf{PH} = \mathbf{P}$.
7. Suppose A is $\mathbf{PSPACE}$-complete. Then $A \in \mathbf{P}$ iff $\mathbf{PSPACE} = \mathbf{P}$.
8. Language $\{\langle G, s, t\rangle : G$ is an undirected graph with a path from s to $t\}$ is in $\mathbf{L}$.
9. Language $\{\langle G, s, t\rangle : G$ is a directed graph with a path from s to $t\}$ is in $\mathbf{NL}$.
10. Language $\{\langle \varphi\rangle \in \textsc{Sat} : \varphi$ has at most one positive literal in each clause$\}$ is $\mathbf{NP}$-complete.
11. Language $\{\langle G\rangle : G$ has an independent set of size $n/2\}$ is $\mathbf{NP}$-complete.
12. Resolution is a polynomially-bounded proof system.
13. The equality problem $\textsc{Eq}_n$ has a 0-fooling set of size 2^n.
14. Certificate complexity is at most sensitivity, that is, $C(f) \leq s(f)$ for all f.
15. Any monotone Karchmer–Wigderson game has a randomised $O(\log n)$-bit protocol.

For each statement, write down one of

- **T** if the statement is known to be true.
- **F** if the statement is false *or not known to be true.* E.g., both $\mathbf{P} = \mathbf{NP}$ and $\mathbf{P} \neq \mathbf{NP}$ should be marked **F**.
- or leave your answer empty.

A correct **T**/**F** answer is worth +1 point, an incorrect answer is worth −1 point, and an empty answer is worth 0 points.

Solution:

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
F	T	T	T	T	F	T	T	T	F	T	F	F	F	F

2 *(15 pts)* **Problem garden.** For each problem below, classify it into the smallest complexity class as you can (as seen in the course). Write also a *short* explanation (one or two sentences).

1. **Input:** A DNF formula φ. **Accept** iff φ is a tautology (that is, $\varphi(x) = 1$ for all x).
2. **Input:** A CNF formula φ. **Accept** iff there is a CNF that is smaller than φ and equivalent to φ (computes the same function).
3. **Input:** Graphs G and H. **Accept** iff G and H are isomorphic.
4. **Input:** Graph G and integer k. **Accept** iff G does not have a vertex cover of size k, but for every node v, deleting v would make it have a vertex cover of size k.
5. **Input:** Graph G. **Accept** iff G is 2-colourable. (Here G is *k-colourable* if there exists a function $c\colon V(G) \to \{1, 2, \ldots, k\}$ such that $c(v) \neq c(u)$ for all $\{v, u\} \in E(G)$.)

Solution:

1. This is **coNP**-complete, ϕ is a tautology iff $\neg\phi$ is in UNSAT.
2. $\Sigma^2_{\mathbf{P}}$: $\exists\phi'$ such that $\forall x\ |\phi'| < |\phi|$ and $\phi(x) = \phi'(x)$.
3. **NP**: the certificate is a bijection from $V(G)$ to $V(H)$ that preserves edges.
4. The language of graphs with no $\leq k$ vertex cover is in **coNP**. The language of graphs G such that all $G - v$ for $v \in V(G)$ have a $\leq k$-size vertex cover is in **NP**: the certificate is just description of these $|V(G)|$ vertex covers. Thus, the target language is in DP. (2 points for identifying it is in $\mathbf{P}^{\mathbf{NP}}$, 1 points for $\Sigma^2_{\mathbf{P}}$ or $\Pi^2_{\mathbf{P}}$)
5. We can reduce this problem to undirected connectivity: consider $H = (V \times \{1, 2\}, \{(v, i), (u, 3-i) \mid uv \in E(G) \wedge i \in \{1, 2\}\})$, then if there is a path from $(u, 1)$ to $(u, 2)$ in H, it means G is not 2-colourable. If there is no such path between any such pair, then the graph is 2-colourable (take any node v in a connected component and colour u to the colour i iff (u, i) is reachable from $(v, 1)$). Therefore the language is in **L**, since undirected connectivity is. (2.5 points for identifying it is in **NL**, 1 point for **P**).

3 *(15 pts)* **Reductions.** Show a reduction 3-Colouring $\leq_p$ Barely-3-Sat where

$$\text{3-Colouring} \coloneqq \{\, \langle G\rangle : G \text{ is 3-colourable} \,\},$$
$$\text{Barely-3-Sat} \coloneqq \{\, \langle \varphi\rangle : \varphi \text{ is a 3-CNF that has a truth assignment that satisfies exactly one literal in each clause} \,\}.$$

Solution: We construct a polynomial-time reduction f. Given a graph G as input, let $f(G)$ be the 3-CNF φ on variables

- v_r, v_g and v_b for every vertex v in G,
- e_r, e_g and e_b for every edge e in G.

The variable v_r should be thought of as an indicator for the event "v is coloured red", similar for green and blue. The variables e_i are dummy variables. The clauses of φ are

- $(v_r \vee v_g \vee v_b)$ for every vertex v in G,
- $(v_c \vee v'_c \vee e_c)$ for every edge $e = \{v, v'\}$ and every colour $c \in \{\text{r,g,b}\}$.

The function f can clearly be computed in polynomial time. We now argue correctness.

If G is 3-colourable, pick any colouring of G with colours $\{\text{r,g,b}\}$ and consider the assignment

- $v_c = 1$ if and only if vertex v is coloured c,
- $e_c = 1$ if and only if the edge e does not touch a vertex of colour c.

Clearly, this assignment satisfies exactly one literal in each clause of φ. Conversely, assume that we have an assignment that satisfies exactly one literal in each clause of φ. For each vertex v in G, there is exactly one colour c such that $v_c = 1$, so lets colour v with colour c. If there were an edge e in G between two vertices of some same colour c, then the associated clause would have at least 2 true literals, a contradiction. We conclude that G is 3-colourable.

Markscheme

5P: Vertex-clause construction

5P: Edge-clause construction

5P: Proof of correctness

4 *(20 pts)* **Polynomial hierarchy.** Show that $\mathbf{NP}^{\text{SAT}} = \Sigma_2\mathbf{P}$.

Solution: The easy direction is to show $\Sigma_2\mathbf{P} \subseteq \mathbf{NP}^{\text{SAT}}$: consider a language L of all x such that $\exists y \forall z\colon f(x, y, z)$ for a poly-time predicate f, which can be computed by a poly-size circuit. Then the non-deterministic TM can guess y and query the SAT-oracle with $\neg f(x, y, z)$, where x and y are hardwired to the circuit for f. Then TM accepts if the answer of the oracle is `no` and rejects otherwise. Clearly $\neg\exists z\colon \neg f(x, y, z) \iff \forall z\colon f(x, y, z)$, so TM accepts x iff there is a witness y such that $\forall z\colon f(x, y, z)$.

The hard direction is to show $\mathbf{NP}^{\text{SAT}} \subseteq \Sigma_2\mathbf{P}$. Consider a SAT-oracle NTM M that accepts a language L. Suppose in the accepting path p it makes SAT-queries $\phi_1, \ldots, \phi_m$ that all have at most m variables (can assume it wlog), and gets answers $(\alpha_1, \ldots, \alpha_m) \in \{0,1\}^m$, let $Y = \{i \in [m] \mid \alpha_i = 1\}$ and $N = [m] \setminus Y$. Then

$$x \in L \iff \exists p, \phi_1, \ldots, \phi_m, \alpha_1, \ldots, \alpha_m \exists z \in (\{0,1\}^m)^Y \forall z' \in (\{0,1\}^m)^N\colon f(x, p, \phi, \alpha, z, z'),$$

where f checks whether M with non-deterministic choices p queries the oracle with $\phi_1, \ldots, \phi_m$ and receives answers $\alpha_1, \ldots, \alpha_m$ (all of this can be done in polynomial time), and checks that $\phi_i(z_i) = 1$ for $i \in Y$ and $\phi_i(z'_i) = 0$ for $i \in N$. Then the predicate

$$\exists z \in (\{0,1\}^m)^Y \forall z' \in (\{0,1\}^m)^N\colon f(x, p, \phi, \alpha, z, z')$$

verifies that the machine follows the nondeterministic path p, queries ϕ and gets answers α, and that those answers are actually correct.

Scoring: 8p for the easy direction and 12p for the hard direction, partial points for ignoring the 0-answers in the hard direction.

5 *(20 pts)* **Communication complexity.** Define $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ so that $f(x,y) = 1$ iff the Hamming distance between x and y is 1, that is, the strings differ in exactly one bit.

5a Show that f has deterministic communication complexity $\Omega(n)$.

5b Show that f has randomised communication complexity $O(1)$. (*Hint: Consider using the public randomness to partition the n input bits into two parts.*)

Solution:

5a As seen in the course, we know that $\mathrm{D}^{cc}(f) \geq \log(\mathrm{part}(f))$. Let M denote the boolean matrix corresponding to f. For every x there are exactly n values for y such that $f(x,y) = 1$ and vice versa. Thus, any 1-monochromatic rectangle can have size at most $n \times n$. But there are $n \cdot 2^n$ many 1-entries in M. It follows that $\mathrm{part}(f) \geq 2^n/n$ and thus $\mathrm{D}^{cc}(f) \geq \Omega(n)$.

Alternatively, for $x, y \in \{0,1\}^{n-1}$, define $x' = x0$ and $y' = y1$. Now we have

$$\mathrm{EQ}_{\text{n-1}}(x,y) = 1 \iff f(x',y')$$

and the upper bound follows from $\mathrm{D}^{cc}(\mathrm{EQ}) \geq n$, which was seen in lectures.

5b Alice and Bob can use the public randomness to pick a subset of indices S uniformly at random (meaning that $\mathbb{P}(i \in S) = 1/2$ for all $i \in [1,n]$). They now run the protocol for EQ on the pair of strings (x_S, y_S) obtained by restricting to indices in S. They also run the protocol for EQ on the complementary parts of the strings $(x_{\overline{S}}, y_{\overline{S}})$. They return 1 if exactly one of these two subroutines returned 1.

By error-boosting, we can assume that the EQ subroutines succeed with high probability. If $f(x,y) = 1$, then the above protocol succeeds with high probability. If $f(x,y) = 0$, then either $x = y$ or x and y differ in at least 2 places. In the former case, the protocol succeeds with high probability, in the latter case there is a good chance (at least $1/2$) that some but not all indices where x and y differ are in S, and thus the protocol succeeds with probability close to $1/2$.

Thus, our protocol succeeds with high probability in the 1 case and with probability roughly $1/2$ in the 0 case. To boost the success probabilities above $2/3$, we can run the protocol twice and give final answer 1 if and only if both runs returned 1. Also, note that since we just use two subroutines of the constant-communication EQ protocol, our protocol has constant communication as well.

Markscheme

a 5P Correctly quoting a useful result

5P Proving the lower bound

b 5P Describing the protocol

5P Proving it correct

6 *(15 pts)* **Computing query complexity.** Design a polynomial-space algorithm that on input a boolean circuit C outputs $\mathrm{D}^{dt}(C)$, where we identify C with the function $\{0,1\}^n \to \{0,1\}$ it computes. That is, the algorithm should output the least depth of a decision tree that computes the same function as C. Can your algorithm also output an optimal tree? If not, why not?

Solution: Let $f(C, n) \coloneqq \mathrm{D}^{dt}(C)$ for C being a Boolean circuit with n inputs $x_1, \ldots, x_n$. Then suppose that the root node of the optimal decision tree queries the input x_i, then $f(C, n) = 1 + \max\{f(C|_{x_i=0}, n-1), f(C|_{x_i=1}, n-1\}$. Hence, since some input is queried in the root, we have

$$f(C, n) = \min_{i \in [n]} \max\{f(C|_{x_i=0}, n-1), f(C|_{x_i=1}, n-1)\}.$$

The base case is: $f(C, 1) = 1$ if C is not a constant, and 0 if it is. Then $f(C, n)$ can be computed by the formula above, the required memory with $|C| = m$ is $M(m, n) \leq m + n + M(m, n-1) \leq n(m+n)$.

Although the size of the optimal tree might be exponential, it can be printed with only polynomial memory. Suppose the output is in the format `root, left subtree, right subtree` then algorithm can first identify the optimal root query (with the recursion above), print the root description, then run the printing recursively for the left subtree, then run the printing algorithm for the right subtree.