MIRI. Computational Complexity. Final exam.

Last name:
Given name:

- Date: Thursday, June 14, 2018
- Time: 8h30 to 10h30
- Grades will be published on Friday, June 15, 2018
- There are eight multiple-choice questions
- Each question has four choices and exactly one correct answer
- A choice is considered correct only if all statements in it are correct
- Choosing wrong answers does not penalize
- **DO NOT** unstaple the sheets

Question 1 The Cook-Levin Theorem states that:

1. SAT is NP-complete.
2. IP = PSPACE.
3. If EXP $\neq$ NEXP, then P $\neq$ NP.
4. TIME(t^2) $\not\subseteq$ TIME(t) for every time-constructible function t.

Question 2 The only difference in the definitions of the complexity classes NP and RP is that:

1. In NP the machine is required to accept every positive instance with non-zero probability, while in RP the machine is required to accept every positive instance with probability at least $1/2$.
2. In NP the underlying model of computation is the non-deterministic Turing machine, while in RP the underlying model of computation is the probabilistic Turing machine, which is very different.
3. In NP the machines are required to halt in polynomial time with probability 1, while in RP the machines are required to halt in polynomial time with probability at least $1/2$.
4. In NP the machines are required to halt in polynomial time with non-zero probability, while in RP the machines are required to halt in polynomial time with probability at least $1/2$.

Question 3 A language $A \subseteq \{0,1\}^*$ is co-NP-complete if and only if its complement $\overline{A} = \{0,1\}^* \setminus A$ satisfies the following condition:

1. it is a subset of some NP-complete language.
2. it is a superset of some NP-complete language.
3. it belongs to NP and every other problem in NP reduces to it by many-one (Karp) polynomial-time reductions.
4. it belongs to NP and some problem in NP reduces to it by many-one (Karp) polynomial-time reductions.

Question 4 Let C be a Boolean circuit with n Boolean inputs and 1 Boolean output; think of it as taking an n-bit natural number a as input and returning one bit $C(a)$ as output. Such a circuit can be thought of as describing the number

$$N_C := \sum_{a=0}^{2^n-1} C(a) \cdot 2^a.$$

The SUCCINCT PRIMALITY problem is this: Given a Boolean circuit C with n Boolean inputs and 1 Boolean output, determine whether N_C is a prime number.

1. The problem is trivial to solve because such an N_C is always an even number.
2. The problem is obviously co-NP-hard because, given a Boolean circuit as in the definition of the problem, we have that there exists $a \in \{0, \ldots, 2^n - 1\}$ satisfying $C(a) \neq 0$ if and only if N_C is composite, and this is just a variant of the CIRCUIT SAT problem, which is NP-complete.
3. The problem is in EXP because deciding the primality of a number can be done in time polynomial in the bit-length of the given number, and the bit-length of N_C is at most exponential in the size of the given circuit C.
4. All the above are incorrect.

Question 5 Representing TRUE by 1 and FALSE by 0, every Boolean formula $F(x_1, \ldots, x_n)$ with variables $x_1, \ldots, x_n$ may be interpreted as computing a polynomial $P_F(x_1, \ldots, x_n)$ defined according to the following recursion:

$$\begin{array}{lcl} F = x_i & \mapsto & P_F = x_i, \\ F = \text{NOT}(G) & \mapsto & P_F = 1 - P_G, \\ F = \text{AND}(G, H) & \mapsto & P_F = P_G \cdot P_H, \\ F = \text{OR}(G, H) & \mapsto & P_F = 1 - (1 - P_G) \cdot (1 - P_H). \end{array}$$

Exactly one of the statements below is correct. Which one?

1. It is possible to check if two given Boolean formulas F and G are equivalent by testing the polynomial identity $P_F - P_G = 0$ in randomized polynomial time, with error probability bounded by $1/3$, by plugging random values $a_1, \ldots, a_n \in \{0, 1\}$ for the variables $x_1, \ldots, x_n$.
2. It is possible to check if two given Boolean formulas F and G are equivalent by testing the polynomial identity $P_F - P_G = 0$ in randomized polynomial time, with error probability bounded by $1/3$, by plugging random values $a_1, \ldots, a_n \in \{0, \ldots, 10 \cdot (|F| + |G|)^2\}$ for the variables $x_1, \ldots, x_n$.
3. The first two statements are wrong because the polynomials constructed this way are exponentially big.
4. The first two statements are wrong because F and G could be equivalent as Boolean formulas without the polynomial identity $P_F - P_G = 0$ being true over the reals.

Question 6 Using the mapping from formulas $F(x_1, \ldots, x_n)$ to polynomials $P_F(x_1, \ldots, x_n)$ given in Question 5, the first step in the sum-check IP protocol to test if F has exactly v satisfying assignments goes as follows: it receives the coefficients $c_0, \ldots, c_d$ of a polynomial $\hat{P}(x_1) = c_0 + c_1 \cdot x_1 + \cdots + c_d \cdot x_1^d$ claimed to agree with the polynomial

$$P(x_1) := \sum_{a_2, \ldots, a_n \in \{0,1\}} P_F(x_1, a_2, \ldots, a_n)$$

and, with arithmetic over the appropriate finite field,

1. checks that $\hat{P}(0) = \hat{P}(1) = v$.
2. checks that $\hat{P}(0) + \hat{P}(1) = v$.
3. checks that $\sum_{b=1}^{10 \cdot |F|^2} \hat{P}(b) = v$.
4. does something that has nothing to do with all the above.

Question 7 Let p be a prime number, let $\mathbb{Z}_p$ denote the field of arithmetic mod p. For each $a, b \in \mathbb{Z}_p$, let $P_{a,b} : \mathbb{Z}_p \to \mathbb{Z}_p$ be the quadratic polynomial defined by $P_{a,b}(x) = ax^2 + bx \bmod p$. Define $\mathcal{H} = \{P_{a,b} : a, b \in \mathbb{Z}_p\}$. Only one is correct:

1. $\mathcal{H}$ is a 2-universal family of functions (also known as pairwise independent family).
2. The first answer is not correct, but if $\mathcal{H}$ were redefined to allow only *non-zero* $a, b \in \mathbb{Z}_p$, then the first answer would be correct.
3. The first answer is incorrect because $\Pr[\mathbf{h}(x) = y] \neq 1/p$ for some $x, y \in \mathbb{Z}_p$, where $\mathbf{h}$ denotes a function taken uniformly at random from $\mathcal{H}$.

4. The first answer cannot be right because 2-universal families of functions must be doubly exponentially big in the size of the domain or in the size of the range, while $|\mathcal{H}| = p^2$ is polynomial in both.

Question 8 Let p be a large prime, much bigger than 2, let $\mathbb{Z}_p$ denote the field of arithmetic mod p. Alice wants to use the computational power of his friends Bob and Charles, who reside in distant galaxies and cannot talk to each other, to compute a linear function $f(x) = ax + b$ (known to all three) on a given input $z \in \mathbb{Z}_p$ (known only to Alice) without actually revealing what z is at all:

- Alice chooses a random $y \in_R \mathbb{Z}_p$ and computes $z_1 = z + y$ and $z_2 = z + 2y$,
- sends z_1 to Bob and asks him to compute $f(z_1)$,
- sends z_2 to Charles and asks him to compute $f(z_2)$,
- gets answers a_1 and a_2,
- interpolates the unique linear function $g(x)$ such that $g(i) = a_i$ for $i = 1, 2$,
- returns $g(0)$.

Only one is correct:

1. The reason this works is that f is linear: had it been a polynomial of degree 2, the task would have been provably impossible even with the help of a third friend David.
2. The reason this reveals nothing about z to Bob and Charles is that z_1 and z_2 are uniformly distributed in $\mathbb{Z}_p$, and although z_1 and z_2 are not uncorrelated because $2z_1 - z_2 = z$, Bob and Charles can't guess anything about z from just knowing their z_i.
3. The reason this reveals nothing about z is precisely the fact that $2z_1 - z_2 = z$.
4. This doesn't work at all: $g(0)$ gives b instead of $az + b$.
