Multiple-choice questions

1. Given a problem $A \in$ NPC. Then it holds that:
   - A. A does not have a polynomial algorithm, unless P = NP.
   - B. yes-instances and no-instances of A can be verified in polynomial time.
   - C. when A is also in coNP, then NP=coNP.
   - D. for every coNP problem B it holds that $B \leq A^c$.

2. Consider three decision problems A, B, and C. It is known that $A \in$ coNP, $B \in$ NP, and $C \in$ NP-hard under $\leq$. Then it follows that:
   - A. If $C \leq B$ then $B \in$ NPC.
   - B. If $C \leq A$, then NP = coNP.
   - C. $A \leq B$ or $B \leq A$.
   - D. $B \leq C$.

3. Which of the following claims is/are true, if any?
   - A. all problems in P$-\{\emptyset, \Sigma^*\}$ are polynomially reducible to each other.
   - B. all PSPACE problems are polynomially reducible to each other.
   - C. all NPC problems are polynomially reducible to each other.
   - D. all NP-hard problems are polynomially reducible to each other.

4. Consider the 3-FALSIFICATION problem, defined as follows: Given a set C of 3-SAT clauses over a set of atoms U, does there exist a truth assignment τ such that τ makes all clauses in C false.

   Professor C.L. Ever has designed an algorithm to solve this problem:

```
function SOLVE(U = {u_1, ..., u_n}, C = {c_1, ..., c_m})
    List ← ∅; i ← 0 ; falsifiable ← true
    while i ≠ m and falsifiable do
        i ← i + 1
        Let c_i = {x_1, x_2, x_3}
        for j ← 1,2,3 do
            if negated(x_j) ∈ List then
                falsifiable ← false
            else
                List ← List ∪ {x_j}
            end if
        end for
    end while
    return falsifiable
end function
```

   Here, the `negated()` call inverts the literal, i.e. a is turned into $\neg a$ and $\neg a$ is turned into a.

   From this algorithm we may conclude that:
   - A. 3-FALSIFICATION $\in$ P.
   - B. The above algorithm only checks whether there are clauses in C in which both a proposition symbol and its negation occur, but it says nothing about the ability to make all clauses false.
   - C. The algorithm is incorrect as 3-FALSIFICATION cannot be solved in polynomial time, unless P = NP.
   - D. 3-FALSIFICATION is the complement of the NP-complete satisfiability problem 3-SAT. Therefore 3-FALSIFICATION is coNP-complete and the algorithm cannot be correct.

12 pt.

12 pt.

4

5. Dr. S. Timation claims:

   It is known that there exists a 2-approximation algorithm for Vertex Cover (VC). Because VC $\in$ NPC, it follows that every NP problem A is reducible to VC. Hence, for every NP-problem there is a 2-approximation algorithm.

   Which of the following hold(s)?

   A. This claim is false, because VC does not have a 2-approximation algorithm.
   B. This claim is false, because VC $\notin$ NPC.
   C. This claim is false, because polynomial time reductions do not preserve approximation ratio's.
   D. This claim holds.

$B \subseteq C$

3

6. Consider the decision problems A, B, and C, where $A \leq B$, and B is a restriction of C. It follows that:
   A. If A is strong NP-complete, then C is also strong NP-complete.
   B. If C has a pseudo-polynomial algorithm, then B also has a pseudo-polynomial algorithm.
   C. If C is NP-complete, then B is NP-complete.
   D. If B is NP-complete, then C is NP-hard.

2

7. The ALARM problem consists of an undirected graph $G = (V, E)$, a distance $d \in \mathcal{Z}^+$ and a budget $B \in \mathcal{Z}^+$. The question is whether there is a subset $V' \subseteq V$ with $|V'| \leq B$, such that every node in $v \in V$ is not more than d units away from a node in V'. I.e., the question is, can we find for every node $v \in V$, a node $v' \in V'$ such that there is a path from v to v' in G with a length of at most d.

   Which of the following claims is/are true?

   A. instances of this problem with $d = 1$ are solvable in polynomial time.
   B. instances of this problem with $d = |V|$ are solvable in polynomial time.
   C. instances of this problem with $d = 1$ are vertex cover instances.
   D. instances of this problem with $B \leq 2$ are solvable in polynomial time.

4

8. Check which of the following statements is true:
   A. If $A \in$ P then $A^c \in$ P $\cap$ NP .
   B. $P^{NP} = P^{coNP}$
   C. If $A \in$ NP then $A^c \in$PSPACE.
   D. NP $\cup$ coNP $\subseteq P^{NP}$.

4

9. M is a polynomial c-approximation algorithm for a minimisation problem A with $c > 1$. Suppose we know that for an instance x of A it holds that $a \leq opt(x) \leq b$ for some constants a and b. Then the result $M(x)$ of applying M to x satisfies:
   A. $a \leq M(x) \leq c$
   B. $M(x) \leq b$
   C. $a \leq A(x) \leq b \times c$
   D. $c/a \leq A(x) \leq c/b$

$A(x) = M(x)$?

4

10. Let L be a language and M a polynomial probabilistic Turing machine accepting L. Which of the following statements allow us to conclude that $L \in$ BPP?
   A. $P[M \text{ accepts } x \mid x \in L] \geq 0.45$ and $P[M \text{ accepts } x \mid x \notin L] \leq 0.55$
   B. $P[M \text{ accepts } x \mid x \in L] \geq 0.55$ and $P[M \text{ accepts } x \mid x \notin L] \leq 0.45$
   C. $P[M \text{ accepts } x \mid x \in L] \geq 0.95$ and $P[M \text{ accepts } x \mid x \notin L] \leq 0.95$
   D. $P[M \text{ accepts } x \mid x \in L] \geq 0.509$ and $P[M \text{ accepts } x \mid x \notin L] \leq 0.499$

33 pt / 40

Lus pt = $50 \times \frac{40 - 1{,}5 \times 7}{40}$ = 36,875 (30 mins)