

Quiz 10

Tick the statements that are true and leave unticked those that are false.
Each statement is worth 1 point for a correct answer.

Your name: _____

- ☒ Every NP-complete language is NP-hard.
- ☐ Every NP-hard language is NP-complete.
- ☒ If A is NP-hard and $A \leq_p B$, then B is NP-hard.
- ☐ If B is NP-hard and $A \leq_p B$, then A is NP-hard.
- ☒ If there is an NP-complete language that belongs to class P, then $P = NP$.
- ☒ If there is an NP-hard language that belongs to class P, then $P = NP$.
- ☒ SAT is NP-complete.
- ☒ CLIQUE is NP-complete.
- ☐ Since $SAT \leq_p CLIQUE$ and $CLIQUE \leq_p SAT$, we have $SAT = CLIQUE$.
- ☒ Every NP-complete language belongs to class EXP.

Bonus. Consider the class $CONP = \{\bar{L} \mid L \in NP\}$, where $\bar{L} = \{w \in \Sigma^* \mid w \notin L\}$. A language L is called CONP-complete if $L \in CONP$ and $A \leq_p L$ for all $A \in CONP$. Give an example of a CONP-complete language. Justify your answer.

Possible answer: The language $\overline{SAT}$ of unsatisfiable Boolean formulas is CONP-complete. Indeed, since $SAT \in NP$, we have $\overline{SAT} \in CONP$. Now take any $A \in CONP$. Then $\bar{A} \in NP$, so the NP-completeness of SAT gives a polynomial-time reduction r such that

$$w \in \bar{A} \iff r(w) \in SAT.$$

For every word w , this gives

$$w \in A \iff w \notin \bar{A} \iff r(w) \notin SAT \iff r(w) \in \overline{SAT}.$$

Thus $A \leq_p \overline{SAT}$.

Explanations

- ☒ *Every NP-complete language is NP-hard.* This follows directly from the definition of NP-completeness: an NP-complete language is NP-hard and also belongs to NP.
- ☐ *Every NP-hard language is NP-complete.* NP-hardness does not require membership in NP. For example, HALT is NP-hard, but it is not in NP.
- ☒ *If A is NP-hard and $A \leq_p B$, then B is NP-hard.* For every $L \in \text{NP}$, we have $L \leq_p A$. By transitivity, $L \leq_p B$, so B is NP-hard.
- ☐ *If B is NP-hard and $A \leq_p B$, then A is NP-hard.* Take $A = \emptyset$ and $B = \text{SAT}$. The constant function that outputs a fixed unsatisfiable formula reduces $\emptyset$ to SAT, but $\emptyset$ is not NP-hard: a yes-instance of SAT cannot be mapped to a member of $\emptyset$.
- ☒ *If there is an NP-complete language that belongs to class P , then $P = \text{NP}$.* Every language in NP reduces to that language, so composing the reduction with its polynomial-time decider puts every language in NP in P . Since $P \subseteq \text{NP}$, the classes are equal.
- ☒ *If there is an NP-hard language that belongs to class P , then $P = \text{NP}$.* The same argument applies: every language in NP reduces to an NP-hard language, whether or not that language is itself in NP.
- ☒ *SAT is NP-complete.* By the Cook–Levin theorem.
- ☒ *CLIQUE is NP-complete.* By another theorem from Lecture 10.
- ☐ *Since $\text{SAT} \leq_p \text{CLIQUE}$ and $\text{CLIQUE} \leq_p \text{SAT}$, we have $\text{SAT} = \text{CLIQUE}$.* Reductions may transform an input into a different encoding. Reductions in both directions show polynomial-time equivalence, not that the two languages contain exactly the same words.
- ☒ *Every NP-complete language belongs to class EXP.* An NP-complete language belongs to NP, and $\text{NP} \subseteq \text{EXP}$.