

Quiz 9

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 language in P belongs to NP, and every language in NP belongs to EXP.
- ☐ A verifier for a language in NP must find a certificate whenever one exists.
- ☒ For $L \in \text{NP}$, if $w \in L$ then there are one or more certificates s such that the verifier accepts (w, s) .
- ☒ For $L \in \text{NP}$, if $w \notin L$ then the verifier rejects (w, s) for every s .
- ☐ If you prove that $\text{P} = \text{NP}$, you must pay \$1 million to the Clay Mathematics Institute.
- ☒ We can say for sure that $\text{HALT} \notin \text{NP}$.
- ☐ If $A \subseteq B$ then $A \leq_p B$.
- ☐ If $A \leq_p B$ then $A \subseteq B$.
- ☒ If $A \leq_p B$ and $B \in \text{P}$, then $A \in \text{P}$.
- ☐ If $A \leq_p B$ and $A \in \text{P}$, then $B \in \text{P}$.

Bonus. Find two languages A and B such that neither $A \leq_p B$ nor $B \leq_p A$. Justify the answer.

Possible answer: Let $A = \emptyset$ and $B = \Sigma^*$. No reduction from A to B can preserve the answer: every output belongs to B , although no input belongs to A . The reverse reduction is impossible for the symmetric reason.

Explanations

- ☒ *Every language in P belongs to NP, and every language in NP belongs to EXP.* A polynomial-time decider is also a verifier: it ignores the certificate. For a language in NP, an exponential-time machine can enumerate all polynomially bounded certificates and run the verifier on each.
- ☐ *A verifier for a language in NP must find a certificate whenever one exists.* The verifier checks a certificate supplied to it; the definition only requires that an accepting certificate exists for each yes-instance.
- ☒ *For $L \in \text{NP}$, if $w \in L$ then there are one or more certificates s such that the verifier accepts (w, s) .* By the definition of class NP.
- ☒ *For $L \in \text{NP}$, if $w \notin L$ then the verifier rejects (w, s) for every s .* By the definition of class NP.
- ☐ *If you prove that $P = \text{NP}$, you must pay \$1 million to the Clay Mathematics Institute.* The statement reverses who pays: the Institute offers the prize for an accepted solution to the P versus NP problem.
- ☒ *We can say for sure that $\text{HALT} \notin \text{NP}$.* Since $\text{NP} \subseteq \text{EXP}$, every language in NP is decidable. The halting problem is undecidable.
- ☐ *If $A \subseteq B$ then $A \leq_p B$.* For example, $\text{HALT} \subseteq \Sigma^*$. But Σ^* is decidable, so $\text{HALT} \leq_p \Sigma^*$ would imply that HALT is decidable (even in polynomial time).
- ☐ *If $A \leq_p B$ then $A \subseteq B$.* For example, over the binary alphabet, $\{0\} \leq_p \{1\}$: send 0 to 1 and every other word to 11. This preserves membership even though $\{0\} \not\subseteq \{1\}$.
- ☒ *If $A \leq_p B$ and $B \in P$, then $A \in P$.* Compute the reduction, then run the polynomial-time decider for B on its output. The reduction runs in polynomial time and produces an output of polynomial length, so the total time is polynomial.
- ☐ *If $A \leq_p B$ and $A \in P$, then $B \in P$.* The language $\emptyset$ is in P, and $\emptyset \leq_p \text{HALT}$: send every input to a fixed non-halting program/input. But HALT is undecidable.

Bonus. Let $A = \emptyset$ and $B = \Sigma^*$. Suppose first that a reduction r from A to B exists. Since $\epsilon \notin A$, answer preservation requires $r(\epsilon) \notin B$. But $B = \Sigma^*$ contains every word, including $r(\epsilon)$, a contradiction. Now suppose that a reduction q from B to A exists. Since $\epsilon \in B$, answer preservation requires $q(\epsilon) \in A$. But $A = \emptyset$, so this is also a contradiction. Thus neither reduction exists.