The completed test is to be submitted electronically to colbourn@asu.edu before the deadline.

Answer each of questions 1-4, one of questions 5 or 6, one of questions 7 or 8, and one of questions 9 or 10. If you answer both questions 5 and 6 (or 7 and 8, or 9 and 10), only the first will be graded. Questions 1–4 are worth 10 points each; questions 5–10 are worth 20 points each. (So you will answer a total of $(4 \times 10) + (3 \times 20) = 100$ points.)

You can use any materials desired, but this is an individual effort, so limit your discussions with fellow students, and under no circumstances should written work be shared.

Part of the exercise of doing the questions is understanding the questions. Read each carefully so as not to waste time.

1. Question 1.

A spanning tree of a connected graph is a connected spanning subgraph that contains no cycles. A leaf in a spanning tree is a vertex of degree one (i.e. in exactly one edge). A spanning tree is cubic if every vertex has degree three or one (that is, all non-leaf vertices appear in exactly three edges). Show that the problem of deciding if a graph has a cubic spanning tree is NP-complete.

2. Question 2.

$A_{LBA} = \{ \langle M, w \rangle : M$ is an LBA that accepts $w \}$. Show that $A_{LBA}$ is PSPACE-complete.

3. Question 3.

(a) Let $L$ be a unary language (a subset of $\{1\}^*$), and suppose that $A \leq_p L$. What can be said about the computability or complexity of $A$?
(b) Let $C$ be a context-free language, and consider the classes $P^C$ and $NP^C$ ($P$ and $NP$ given an oracle for $C$). How do these classes relate to $P$ and $NP$? Can you conclude anything about whether or not $P^C = NP^C$?

(Here \( L = SPACE(\log n) \).) Suppose that \( A \in L \) and \( B \in L \). Must \( A \cap B \) be in \( L \)? Must \( A \cup B \) be in \( L \)? Must \( AB \) be in \( L \)? Explain.

5. Question 5.

A narcissistic Turing machine is a TM \( M \) for which \( L(M) = \{ \langle M \rangle \} \).

(a) Show that a narcissistic TM exists.
(b) Determine whether or not the language of (encodings of) narcissistic Turing-machines is Turing-recognizable. Also determine whether or not the language of (encodings of) narcissistic Turing-machines is co-Turing-recognizable. Explain carefully.


(a) Show that \( A_{\text{DFA}} = \{ \langle M, w \rangle : M \text{ is a DFA that accepts } w \} \) is in \( SPACE(\log n) \).
(b) Show that \( A_{\text{NFA}} = \{ \langle M, w \rangle : M \text{ is an NFA that does not accept } w \} \) is in \( NSPACE(\log n) \) (NL).
(c) Show that if \( BIN_{\text{REX}} = \{ \langle R \rangle : \{0,1\}^* \subseteq L(R) \} \) is in P, then \( P=NP \).

7. Question 7.

Let \( \mathbb{N} \) be the set of natural numbers, and define the relation \( \uparrow \), a subset of \( \mathbb{N}^3 \), so that \( (x, y, z) \) are related by \( \uparrow \) exactly when \( z = \max(x, y) \). We consider the theory of \( \mathbb{N} \) and \( \uparrow \).

(a) Show that \( x = y \) and \( z = \min(x, y) \) are definable in this theory.
(b) Show that the theory of \( \mathbb{N} \) and \( \uparrow \) is decidable.
(c) Show in detail that the distributive property

\[
\max(\min(a, b), c) = \min(\max(a, c), \max(b, c))
\]

holds using the decidability of the theory.

8. Question 8.

(a) Show that \( A_{\text{NFA}} = \{ \langle M, w \rangle : M \text{ is a NFA that accepts } w \} \) is NL-hard.
(b) Show that \( INFINITE_{\text{DFA}} = \{ \langle M \rangle : M \text{ is a DFA with } L(M) \text{ infinite} \} \) is NL-complete.
(c) Show that \( EMPTY_{\text{DFA}} = \{ \langle M \rangle : M \text{ is a DFA with } L(M) \text{ empty} \} \) is NL-complete.

We do not know whether NP = coNP or not. Prove that there are oracles \( A \) and \( B \) for which \( NP^A = coNP^A \) and \( NP^B \neq coNP^B \).

10. Question 10.

(a) A type I log space verifier is a (deterministic) TM with a read-only input tape, a read-only certificate tape, and a read-write work tape. All three tape heads can move left or right, while staying on the tape; and the work tape has \( O(\log n) \) cells, where \( n \) is the size of the input. Show that the set of languages with type I log space verifiers is NP.

(b) A type II log space verifier is a (deterministic) TM with a read-only input tape, a read-only certificate tape, and a read-write work tape. Input and work tape heads can move left or right, while staying on the tape; certificate tape head can only move right; and the work tape has \( O(\log n) \) cells, where \( n \) is the size of the input. Show that the set of languages with type II log space verifiers is NL.