V22.0453-001: Honors Theory of Computation

Final Exam

Instructions: Solve all the problems. You can assume all the results taught in class.

Due on Thu, Dec 16, at 9:30am (slide under office door (# 416)).

Here are definitions of standard complexity classes:

\[ L = \text{DSPACE}(O(\log n)) \]
\[ NL = \text{NSPACE}(O(\log n)) \]
\[ P = \bigcup_{k=1,2,...} \text{DTIME}(n^k) \]
\[ NP = \bigcup_{k=1,2,...} \text{NTIME}(n^k) \]
\[ PSPACE = \bigcup_{k=1,2,...} \text{DSPACE}(n^k) \]
Problem 1: [15 Points]
For each of the following statements, answer whether it is true or false with a brief explanation. No credit will be given without an explanation.

1. There is a language in PSPACE that is not in L.
2. The following language is decidable:
   \[ A = \{ \langle \phi \rangle \mid \phi \text{ is a boolean formula that has exactly one satisfying assignment} \}. \]
3. Let \( f : \{0,1\}^n \to \{0,1\} \) be a boolean function defined as follows:
   \[ f(x) = 1 \text{ if and only if } \text{ the string } x \text{ has more 1's than 0's.} \]
   Then \( f \) can be computed by a circuit of polynomial (in \( n \)) size.

Problem 2: [20 Points]
Show that between the following two languages, \( L_1 \) is decidable and \( L_2 \) is undecidable.

1. \( L_1 = \{ \langle M \rangle : M \text{ is a TM and } M \text{ takes more than 453 steps on some input} \} \).
   \text{Hint: In } k \text{ steps, the tape head can access at most first } k \text{ cells of the input.}
2. \( L_2 = \{ \langle M \rangle : M \text{ is a TM and } M \text{ takes more than 453 steps on some input in } L(M) \} \).
   \text{Hint: Use a reduction from the undecidable language } \{ \langle M' \rangle \mid M' \text{ accepts } \varepsilon \}.

Problem 3: [15 Points]
A graph \( G \) is called bipartite if its set of vertices can be partitioned into two sets \( A \) and \( B \) such that every edge has one endpoint in \( A \) and one endpoint in \( B \). Let
\[ \text{NON-BIPARTITE} = \{ \langle G \rangle \mid G \text{ is not bipartite} \}. \]

- Show that a graph is non-bipartite if and only if it has a cycle of odd length.
- Show that NON-BIPARTITE \( \in \) NL.

Problem 4: [10 Points]
Recall that a vertex cover in a graph \( G \) is a subset \( S \) of vertices such that for every edge \( e \) in the graph, at least one of the endpoints of \( e \) is contained in \( S \). Consider the following decision problem:
\[ \text{VERTEX-COVER} = \{ \langle G, k \rangle \mid G \text{ has a vertex cover of size } k \}. \]
If \( P=NP \), show that there is a polynomial time algorithm that given an \( n \)-vertex graph \( G \) and an integer \( k \), \( 1 \leq k \leq n \), finds a vertex cover of size \( k \) if one exists and outputs NO otherwise.

\text{Hint: VERTEX-COVER} \( \in \) NP and if \( P=NP \), then it can be decided in polynomial time by some algorithm. Show how one can use this algorithm to actually find a vertex cover of size \( k \) (if it exists) in polynomial time.
Problem 5: [10 Points]
Let FACTOR be the following language:

$$\text{FACTOR} = \{ \langle n, a, b \rangle \mid n, a, b \text{ are positive integers represented in binary such that } \exists d, a \leq d \leq b, \text{ and } d \text{ divides } n \}.$$ 

I.e. $\langle n, a, b \rangle \in \text{FACTOR}$ if and only if $n$ has an integer divisor in the range $[a, b]$.

• Show that $\text{FACTOR} \in \text{NP}$.

• Show that if $\text{P}=\text{NP}$, one can factorize integers in polynomial time, i.e. find a complete factorization of given integer $n$ in time poly$(\log n)$ (Hint: Use binary search).

Problem 6: [10 Points]
Let $\Sigma = \{(, )\}$ and BALANCED be the language over alphabet $\Sigma$ consisting of all strings with balanced parentheses. For example, the following strings are in BALANCED

$$( (), ()(), ()(((())())$$

whereas following strings are not in BALANCED

$$((()) (())() )()$$

• Show that BALANCED is a context free language. Give both: a grammar as well as a pushdown automaton.

• Show that BALANCED $\in \text{L}$. 