6.045J/18.400J: Automata, Computability and Complexity Nancy Lynch

Homework 12.1 (Fake)

Due: Never *Elena Grigorescu*

Readings: Sipser, Chapter 8 (the whole chapter).

Problem 1: (Sipser Exercise 8.1) Show that for any function $f : N \rightarrow N$, where $f(n) \geq n$, the space complexity class SPACE($f(n)$) is the same whether you define the class by using the single-tape TM model or the two tape read-only TM model.

Problem 2: (Sipser Problem 8.10) The Japanese game *go-moku* is played by two players, "X" and "O", on a 19×19 grid. Players take turns placing markers, and the first player to achieve 5 of his/her markers consecutively in a row, column, or diagonal, is the winner. Consider this game generalized to an $n \times n$ board. Let

$$\text{GM} = \{\langle P \rangle |\ P \text{ is a position in generalized go-moku, where player "X" has a winning strategy}\}.$$

By a *position* we mean a board with markers placed on it, such as may occur in the middle of a play of the game. Show that GM $\in$ PSPACE.

Problem 3: The proof of Savitch's theorem, in Section 8.1, describes in general how one can simulate any $f(n)$-space-bounded nondeterministic Turing machine N with an $f^2(n)$-space-bounded deterministic Turing machine M. The key is a recursive computation of the CANYIELD relation, which reuses space.

Give a good upper bound on the *running time* of M on input w.

Problem 4: (Sipser Problem 8.12) Show that TQBF restricted to formulas where the part following the quantifiers is in conjunctive normal form is still PSPACE-complete.

Problem 5: (Sipser Problem 8.17) Let A be the language of properly nested parentheses. For example, (()) and (()(()))() are in A, but)(is not. Show that A is in L.

Problem 6: Show:

1. $A \leq_L B \Rightarrow \bar{A} \leq_L \bar{B}$.
2. $A \leq_L B$ and $B \in$ NL $\Rightarrow A \in$ NL.
3. $A \leq_L B$ and $B \leq_L C \Rightarrow A \leq_L C$.

Problem 7: (Sipser 8.27) Recall that a directed graph is *strongly connected* if every two nodes are connected by a directed path in each direction. Let

$$\text{STRONGLY-CONNECTED} = \{\langle G \rangle |\ G \text{ is a strongly connected graph}\}.$$

Show that STRONGLY-CONNECTED is NL-complete.

Problem 8: This problem uses the ideas in the proof of Theorem 8.27.

Describe a nondeterministic log-space Turing machine M that decides the language

$$L = \{\langle G, s, m, k \rangle |\ G \text{ is a directed graph, } s \text{ is a node in } G,\ m, k \in \mathbb{N}, \text{ and exactly}$$
$$m \text{ nodes of } G \text{ are reachable from } s \in G \text{ by paths consisting of at most } k \text{ edges}\}.$$

That is, if exactly m nodes are reachable from $s \in G$ by paths of length at most k, than M must accept $\langle G, s, m, k \rangle$ on some computation path. On the other hand, if more or fewer than m nodes are reachable from $s \in G$ by paths of length at most k, then M must reject $\langle G, s, m, k \rangle$ on all computation paths.

Explain why M works correctly and why it works in log space.