CSE 531: Computability and Complexity Autumn 2002
Problem Set #4 Instructor: Venkatesan Guruswami
Due on Tuesday, **November 26, 2002** in class.

Instructions: You are permitted (though not exactly encouraged) to collaborate with fellow students taking the class in solving problem sets. If you do so, *please indicate for each problem the people you worked with on that problem.* Note that you must write down solutions on your own and collaboration must be restricted to a discussion of solution ideas. Collaboration is **not allowed** for the optional problem. Solutions are expected to be your original work and so you must refrain from looking up solutions or solution ideas from websites or other literature.

---

1. We showed in class that ALL_{NFA} is in PSPACE. Prove that ALL_{NFA} is PSPACE-complete by giving a polynomial time mapping reduction from every language in PSPACE to ALL_{NFA}.

2. The product of two $n \times n$ Boolean matrices A and B is an $n \times n$ Boolean matrix C where $C_{i,j} = \bigvee_{k=1}^{n} (A_{i,k} \wedge B_{k,j})$.

   (a) Show that matrix multiplication can be performed in logarithmic space (i.e. give a logspace transducer that takes as input A, B and outputs their product $A \cdot B$).

   (b) Prove that computing the p'th power of an $n \times n$ Boolean matrix A can be done in space $O(\log n \log p)$.

   (c) Using the above, give an alternative proof of the inclusion $\text{NL} \subseteq \text{SPACE}(\log^2 n)$.

3. Recall the language 3COLOR consisting of 3-colorable graphs which you proved NP-complete in Problem 4(b) of PS #3. We now consider the complexity of the language of graphs which can colored with just *two* colors. Let us define

$$\text{2COLOR} = \{\langle G \rangle \mid G \text{ is 2-colorable}\} \; .$$

   In this problem, you will show that the complexity of 2COLOR is closely related to that of another fundamental problem: undirected connectivity. Specifically, define

$$\text{UPATH} = \{\langle G, s, t \rangle \mid G \text{ is an } \textit{undirected} \text{ graph that has an undirected path from } s \text{ to } t\} \; .$$

   While PATH (on directed graphs) is the canonical NL-complete problem, the exact space complexity of UPATH is unknown. In particular it is not known whether UPATH is in L, and neither is it known to NL-complete (though it is considered unlikely that UPATH could be NL-complete). Since UPATH is such a fundamental problem which does not fit as a complete problem for the natural complexity classes that we have seen so far, a complexity class is defined expressly for the purpose of capturing its complexity!

   Formally, define SL (stands for "symmetric logspace") to be class of languages that are logspace reducible to UPATH. Call a language A to be SL-complete if: (i) $A \in \text{SL}$, and (ii) $\forall B \in \text{SL}$, $B \leq_L A$. Note that just by its definition, UPATH is SL-complete!

   (a) Prove that $\overline{\text{2COLOR}}$, the complement of 2COLOR, is SL-complete. (Do not forget to show that $\overline{\text{2COLOR}} \in \text{SL}$.)

   (b) Show that $\text{2COLOR} \in \text{NL}$.

4. Problem 8.11, Sipser's book (Properly nested parantheses is in L)

5. Problem 8.17, Sipser's book (Give an example of an NL-complete context-free language)

6. Following the characterization of NP as problems whose solutions can be verified in P with the help of a certificate, we can imagine that perhaps NL can be characterized as the class of logspace verifiable languages defined as follows. Define a language A to be logspace verifiable if there exists a relation $R \subseteq \Sigma^* \times \Sigma^*$ and an integer k such that

   - $x \in A \iff \exists y$ s.t. $\langle x, y\rangle \in R$
   - $\langle x, y\rangle \in R \implies |y| \leq |x|^k$, and
   - $R \in$ L.

   However, this does not (or rather is unlikely to) characterize NL. Prove that, in fact, the class of logspace verifiable languages is exactly NP.

7. * **(Optional Problem)** The algorithm for deciding membership in the language $E_{\text{CFG}} = \{\langle G\rangle | G \text{ is a context-free grammar and } L(G) = \emptyset\}$ corresponding to the emptiness problem for context-free grammars has been quite useful as a subroutine in at least two problems from earlier problem sets (Problem 5 in PS#1 and Problem 6 in PS#2). It is easy to check this algorithm for E_{CFG} (from Theorem 4.7 of the book) runs in polynomial time, so in fact $E_{\text{CFG}} \in P$.

   (a) "The language E_{CFG} in some sense captures the essence of *efficient* solvability" — give a formal basis for this statement by defining a meaningful notion of completeness for the class P and proving that E_{CFG} is P-complete under this notion.

   (b) Recall that we showed that $A_{\text{CFG}} = \{\langle G, w\rangle \mid G \text{ is a grammar that generates } w\}$ is in P using dynamic programming. Prove that A_{CFG} is P-complete according to your definition of P-completeness in (a).
   (Depending upon your reduction, solving this part before (a) might help.)