CSE 531: Computability and Complexity — Autumn 2003

Problem Set #4 — Instructor: Venkatesan Guruswami

Due on Wednesday, **November 26, 2003** in class.

Instructions: Same as Problem Set 1.
There are **SEVEN** problems, including **ONE** *optional* problem.

---

1. Problem 7.34, Sipser's book. (3COLOR is NP-complete)

2. In the previous problem set, you showed that the language
$$2SAT = \{\langle \phi \rangle \mid \phi \text{ is a satisfiable 2CNF formula}\}$$
is in P. You are now required to show that 2SAT is NL-complete.

3. We showed in class that ALL_{NFA} is in PSPACE. In this problem, you are required to prove that ALL_{NFA} is in fact PSPACE-complete.

4. Problem 1 above asked you to show that the language 3COLOR consisting of 3-colorable graphs is NP-complete. 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) $A \in$ SL, and (ii) $\forall B \in$ SL, $B \leq_L A$. Note that just by its definition, UPATH is SL-complete!

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

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

6. This problem concerns ***branching programs*** which are described in Section 10.2 of Sipser's book. We briefly repeat the definition here. A branching program is a directed acyclic graph where all nodes are labeled by variables, except for two output nodes labeled 0 or 1. The nodes that are labeled by variables are called query nodes, each of which has two outgoing

edges, onelabeled 0 and the other labeled 1. Both output nodes have no outgoing edges, and one of the nodes of the branching program is designated the start node. A branching program determines a Boolean function as follows. Take any assignment to the variables appearing on its query nodes and, beginning at the start node, follow the path determined by taking the outgoing edge from each query node according to the value assigned to the indicated variable (i.e. take the 0-edge is the variable is 0 and 1-node if it is 1). Do this until one of the output nodes is reached. The label of this output node is the output of the branching program on that input.

Define a ***family of branching programs*** $\mathcal{B} = (B_1, B_2, B_3, \dots)$ to be an infinite list of branching programs. The n'th member B_n of the list is a branching program that has n input variables $x_1, \dots, x_n$. Say that a family of branching programs ***decides a language*** $A \subseteq \{0,1\}^*$ if for every string a of some length j, $a \in A$ iff $B_j(a) = 1$. Here $B_j(a)$ denotes the output of the branching program B_j when its j input variables $x_1, \dots, x_j$ are set to the values $a_1, \dots, a_j$.

Define the size of a branching program to be the number of nodes in it.

(a) Give a diagram representing the n'th branching program for even n in a family deciding the language $\{w \mid w \in \{0,1\}^* \text{ and } w \text{ has an odd number of 1's}\}$. Your branching program should have size $O(n)$ to receive full credit.

(b) Show that if A is a language in LOGSPACE, then A is decided by a family of branching programs where the n'th member of the family has at most poly(n) nodes.

7. * **(Optional Problem)** Define the language

$$\text{FOREST} = \{\langle G \rangle \mid G \text{ is a forest, i.e., an } \textbf{undirected} \text{ graph with no cycles}\} .$$

Prove that FOREST $\in$ LOGSPACE.