Theory of Computation

Midterm Examination on October 24, 2017
Fall Semester, 2017

Problem 1 (25 points) Prove that E is closed under linear-time reductions. (A class C is closed under linear-time reductions if whenever L is reducible to L' and $L' \in C$, then $L \in C$.)

Proof: Suppose L is reducible to L' by a linear reduction and $L' \in \mathrm{E}$. We proceed to prove that $L \in \mathrm{E}$. Let R be a linear-time reduction from L to $L' \in \mathrm{TIME}(2^{kn})$ for some positive integer k. By definition, $x \in L$ if and only if $R(x) \in L'$. But $|\, R(x) \,| = O(|\, x \,|)$ as R runs in linear time. So "$R(x) \in L'$?" can be answered in time $2^{O(k|\, x \,|)}$. ■

Problem 2 (25 points) Let M be a Turing machine. Define a language

$$\text{PALINDROMES} = \{M \mid M \text{ accepts strings which are palindromes.}\}.$$

Prove or disprove that PALINDROMES is decidable. (Recall that a palindrome is a string which reads the same backward as forward.)

Proof: We disprove that PALINDROMES is decidable. The property of accepting palindromes is nontrivial because the set of Turing machines that accept palindromes is a proper subset of all RE languages. By Rice's theorem, PALINDROMES is undecidable. ■

Problem 3 (25 points) Please answer the following questions:

1. **(5 points)** Let M be a deterministic Turing machine and L a language. Give the formal definitions of:
   (a) M deciding L.
   (b) M accepting L.
2. **(5 points)** Give the definition of recursive language and recursively enumerable language.
3. **(10 points)** Prove that if L is recursively enumerable but not recursive, then $\overline{L}$ is not recursively enumerable.
4. **(5 points)** In the definition of reduction from problem A to problem B, the reduction is required to run within log space or at most polynomial time. Why?

Proof:

1. Let M be a deterministic Turing machine, L a language and x a string:
   (a) We say that M decides L if $M(x) =$ "yes" when $x \in L$ and $M(x) =$ "no" when $x \notin L$.
   (b) We say that M accepts L if $M(x) =$ "yes" when $x \in L$ and $M(x) = \nearrow$ if $x \notin L$.
2. L is a recursive language if there is a Turing machine M that decides L. L is a recursively enumerable language if there is a Turing machine M that accepts L.
3. Assume that $\overline{L}$ is recursively enumerable. Then both L and $\overline{L}$ are recursively enumerable, so by Kleene's theorem (page 154 of the slides) we conclude that L is recursive, a contradiction.
4. To ensure that B is indeed at least as hard as A, the bulk of the computation should be carried out by B, not the reduction. Otherwise, one can hide the complexity of A in the reduction, foiling our plan of establishing the said relative difficulties between two problems.

■

Problem 4 (25 points) Consider the following boolean functions

$$f : \{\, \texttt{true}, \texttt{false} \,\}^n \to \{\, \texttt{true}, \texttt{false} \,\}^m.$$

How many such functions are there? (Use parentheses to avoid ambiguity.)

Proof: $2^{(m(2^n))}$. ■