Theory of Computation

Midterm Examination on November 10, 2015

Fall Semester, 2015

Problem 1 (20 points) Prove that the halting problem H is complete for RE (the set of recursively enumerable languages). (Recall that a problem A is complete for RE if every language in RE can be reduced to A.)

Proof: Let L be any recursively enumerable language. Assume M accepts L. Clearly, one can decide whether $x \in L$ by asking if $M : x \in H$. This reduction is clearly computable. Hence all recursively enumerable languages are reducible to H! ■

Problem 2 (20 points) Let $P(x, y)$ be a binary predicate, and let Q be the unary predicate defined by $Q(a) \Leftrightarrow \neg P(a, a)$. Show that Q is distinct from all the predicates P_b, defined by $P_b(a) \Leftrightarrow P(a, b)$.

Proof: If Q is P_b, then

$$P(b, b) \Leftrightarrow P_b(b) \Leftrightarrow Q(b) \Leftrightarrow \neg P(b, b).$$

■

Problem 3 (20 points) If the following language L is decidable, please give an algorithm; otherwise, prove that it is undecidable by reduction:

$$L = \{M \mid M \text{ is a Turing machine and there exists an input whose length is less than } |M| \text{ on which } M \text{ halts}\}.$$

Proof: L is undecidable. We reduce the halting problem H to L. Given an instance $M; x$, we construct the following TM M' with an arbitrary input y:

$$M'(y) = \begin{cases} \text{yes}, & \text{if } M(x) \neq \nearrow, \\ \nearrow, & \text{otherwise.} \end{cases}$$

For any input y, M' halts at "yes" if and only if M halts on x. In other words, M' halts for all inputs including those of length less than $|M'|$ if and only if M halts on x. So $M' \in L$ if and only if $M; x \in H$. Hence, L is undecidable. ■

Problem 4 (20 points)

1. **(10 points)** Give the definitions of

(a) The complement of a complexity class.

(b) Being closed under complements.

2. **(10 points)** Show that if NP $\neq$ coNP, then P $\neq$ NP. (Half of the grade will be deducted if any of (a) and (b) above is wrongly answered.)

Proof:

1. For the definitions:

   (a) For any complexity class $\mathcal{C}$, co$\mathcal{C}$ is defined as

   $$\text{co}\mathcal{C} = \left\{ L : \overline{L} \in \mathcal{C} \right\}.$$

   (b) We say that a complexity class $\mathcal{C}$ is closed under complement if $\mathcal{C} = \text{co}\mathcal{C}$.

2. P is closed under complementation. If P = NP, then NP is also closed under complementation. In other words, NP = coNP.

■

Problem 5 (20 points) Recall that NL = NSPACE (log n) and REACHABILITY $\in$ NL. Prove that REACHABILITY is NL-complete.

Proof: Let $L \in$ NL be decided by a log-space NTM M. We proceed to prove that REACHABILITY is NL-hard by reducing L to REACHABILITY. Given input x, construct the polynomial-sized configuration graph G of M on input x (see p. 243 of the slides). Note that the nodes represent all configurations of $M(x)$ and the edges represent legal transitions between configurations. Particularly, the START node and the ACCEPT node denote the starting configuration and the accepting configuration, respectively. G is represented by the adjacency matrix which can be generated by the following procedure:

```
for each configuration i do
    for each configuration j do
        if there is a legal transition between i and j then
            Output 1;
        else
            Output 0;
        end if
    end for
end for
Output START, ACCEPT;
```

The reduction can be done in $O(\log)$ space because i and j are encoded in binary. Clearly, $x \in L$ if and only if $R(x) \in$ REACHABILITY. So REACHABILITY is NL-complete. ■