V22.0453-001: Honors Theory of Computation

Problem Set 5

Due Dec 9, 2005

All problems are worth 10 points.

Problem 1

Let L be a Turing-recognizable language over alphabet Σ. Show that there exists a Turing machine M with the following properties:

- The machine has one input tape (as usual) and one *output tape*. The output tape is write-only, meaning, once the tape-head writes a symbol on it, the head can move only to the right. The output tape is empty when the machines starts.
- When the machine is run on empty input (it may never halt), the contents of the output tape are

$$w_1\#w_2\#w_3\#w_4\#\ldots\ldots$$

  Here $w_i \in \Sigma^* \ \forall i$ and $\# \notin \Sigma$ is a special separator symbol.
- $w_i \in L \ \forall i$, and every $w \in L$ appears as some w_i in the list (however the same string could occur multiple times).

Problem 2

Let U be a set. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a collection of subsets of U (i.e. each S_i is a subset of U) such that $U = \bigcup_{i=1}^m S_i$. A subcollection of subsets $\{S_{i_1}, \ldots, S_{i_k}\}$ in $\mathcal{S}$ is a *set cover of size* k if $U = \bigcup_{j=1}^k S_{i_j}$. Define

$$\text{Set-Cover} \quad = \quad \{\langle \mathcal{S}, k\rangle : \mathcal{S} \text{ has a set cover of size } k\}.$$

By reduction from Vertex-Cover, prove that Set-Cover is **NP-Complete**.

Problem 3

1. Prove that if $\mathbf{P} = \mathbf{NP}$, that is, if there is a polynomial-time algorithm that decides 3-SAT, then there is a polynomial-time algorithm that given a 3-SAT instance φ (i.e. a formula in 3-CNF), finds a satisfying assignment to φ if $\varphi \in$ 3-SAT, or outputs NO if $\varphi \notin$ 3-SAT.
2. Prove that if $\mathbf{P} = \mathbf{NP}$, then there is a polynomial-time algorithm that given a 3-SAT instance φ, finds an assignment that satisfies the maximum number of clauses in φ that are satisfiable.

Problem 4

Knapsack is the following problem: We are given n objects, where object i has volume v_i and cost c_i. We also have a bag which has total volume B, and a target cost t. We want to decide whether it is possible to fit in the bag a subset of the objects of total cost at least t. In other words, **Knapsack** is the problem that given volumes $v_1, \ldots, v_n$, costs $c_1, \ldots, c_n$, a volume bound B and a target cost t, each of which is an integer, decide whether there is a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} c_i \geq t$ and $\sum_{i \in S} v_i \leq B$.

Prove, by reduction from **Subset-Sum**, that **Knapsack** is **NP-Complete**.

Problem 5

Partition is the following problem: Given a sequence of n integers $a_1, \ldots, a_n$, decide whether there is a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = \sum_{j \notin S} a_i = \frac{1}{2} \cdot \sum_{i=1}^{n} a_i$.

Prove, by reduction from **Subset-Sum**, that **Partition** is **NP-Complete**.

Problem 6

Bin-Packing is the following problem: Given volumes $v_1, \ldots, v_n$, a volume bound B, and a target number k, each of which is an integer, decide whether we can partition $v_1, \ldots, v_n$ into k disjoint subsets such that the volumes in each subset sum up to at most B.

Prove, by reduction from **Partition**, that **Bin-Packing** is **NP-Complete**.