Written Qualifying Exam
Theory of Computing

Spring 2001

Friday, May 18, 2001

This is a *four hour* examination. All questions carry the same weight. Answer all of the following *six* questions.

• Please check to see that your name and address are correct as printed on your blue-card.
• Please *print* your name on each exam booklet. Answer each question in a *separate* booklet, and *number* each booklet according to the question.

Read the questions carefully. Keep your answers legible, and brief but precise. Assume standard results, except where asked to prove them.

Problem. 1 New booklet please. [10 points]

(a) **(5 points)** Let L_1 and L_2 be two languages. The *shuffle* of L_1 and L_2 is the language $L_1 \# L_2 = \{w_1 \dots w_k \mid k \geq 0$ and each string w_i belongs to L_1 or $L_2\}$. Show that regular languages are closed under the shuffle operation. Moreover, given NFA's for L_1 and L_2, construct an NFA for $L_1 \# L_2$.

(b) **(5 points)** Let the alphabet $\Sigma = \{A, T, C, G\}$. Define the complementation operation on letters of Σ by $A^C = T$, $T^C = A$, $C^C = G$, and $G^C = C$. Given $\sigma = x_1 \dots x_k$, we let $\sigma^C = x_1^C \dots x_k^C$, while $\sigma^R = x_k \dots x_1$. Then we let the language of reverse palindromes $\mathrm{RPAL}(\Sigma)$ over $\{A, T, C, G\}$ to be

$$\mathrm{RPAL}(\{A, T, C, G\}) = \{\sigma \in \{A, T, C, G\}^* \mid \sigma^R = \sigma^C\}.$$

Show that the language $\mathrm{RPAL}(\Sigma)$ is context-free but not regular.

Problem. 2 New booklet please. [10 points]

Given a system of polynomial equations in many variables over integers, Hilbert's 10^{th} problem asks if the system has an integer solution. It was a celebrated result that Hilbert's 10^{th} problem is undecidable. Show a simpler result, namely that the problem is NP-hard. Moreover, show that NP-hardness holds even if all equations have degree at most 6 (if you cannot enforce degree at most 6, get partial credit for general NP-hardness). Is this problem NP-complete? Does your reduction show the NP-hardness of polynomial equations over over the space of real and/or complex numbers instead of integers?

Problem. 3 New booklet please. [10 points] Show that it is undecidable if a Turing Machine with alphabet $\{0, 1, B\}$ ever prints three consecutive 1's on its tape.

Please turn over.

Problem. 4 New booklet please. [10 points]

Let T be a binary tree. For each node v in T, let $h(v)$ be the length of the longest downward path from v to a leaf, and let $d(v)$ be the length of the shortest downward path from v to a leaf. (Thus at a leaf v, $d(v) = h(v) = 0$.) Define a *whole* binary tree to be one in which every internal node has two children, and in addition, $h(v) \leq 2 \cdot d(v)$ for all nodes v.

Let $T(\ell)$ be the smallest possible number of nodes in a whole binary tree with $h(root) = \ell$. Determine $T(\ell)$ exactly.

Hint: The form of the answer may depend on the parity of ℓ, i.e. whether $\ell =$ even or odd.

Problem. 5 New booklet please. [10 points]

The input is a sequence of n elements x_1, x_2, $\ldots$, x_n that we can read *sequentially*. We want to use a memory that can only store $O(k)$ (e.g., $\leq 4k$) elements at a time. Give a high level description of an algorithm that finds the kth *smallest element* in $O(n)$ time.

Hint: Use the linear-time median algorithm.

Problem. 6 New booklet please. [10 points]

Define a *common subsequence* (not necessarily contiguous) of two strings $V = v_1 \cdots v_n$ and $W = w_1 \cdots w_m$ as a pair of sequence of indices:

$$1 \leq i_1 < \cdots < i_k \leq n \quad \text{and} \quad 1 \leq j_1 < \cdots < j_k \leq m,$$

such that

$$\forall_{1 \leq t \leq k} v_{i_t} = w_{j_t}.$$

Let $s(V, W) = k$ be the length of a *longest common subsequence* (LCS) of V and W. For example, the LCS of two strings $V = ATCTGAT$ and $W = TGCATA$ is $TCTA$ and $s(V, W) = 4$. Devise an efficient algorithm to compute $s(V, W)$.