Written Exam

10:00 – 12:30, February 2, 2016

Entrance Examination (AY 2016)

Department of Computer Science
Graduate School of Information Science and Technology
The University of Tokyo

Notice:

(1) Do not open this problem booklet until the start of the examination is announced.

(2) Answer the following 4 problems. Use the designated answer sheet for each problem.

(3) Do not take the problem booklet or any answer sheet out of the examination room.

Write your examinee's number in the box below.

Examinee's number	No.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

Problem 1

A *unit lower triangular matrix* is a lower triangular matrix whose diagonal elements are all equal to 1.

Answer the following questions.

(1) Suppose that L and L' are $n \times n$ lower triangular matrices. Prove that the product of them, LL', is also a lower triangular matrix.

(2) Suppose that L and L' are $n \times n$ unit lower triangular matrices. Prove that LL' is also a unit lower triangular matrix.

(3) Compute the inverse matrices of

$$L_1 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \quad \text{and} \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix},$$

respectively.

(4) Suppose that an $n \times n$ invertible matrix A is decomposed in two ways as $A = LU = L'U'$, where U and U' are upper triangular matrices, and L and L' are unit lower triangular matrices. Prove that $L = L'$ and $U = U'$.

You can use the following facts.

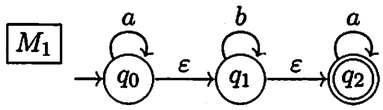
(i) The inverse of an upper triangular matrix, if it exists, is also an upper triangular matrix.

(ii) The inverse of a unit lower triangular matrix always exists, and it is also a unit lower triangular matrix.

Problem 2

Let us consider *nondeterministic finite automata with ε-transitions (ε-NFAs)* over the alphabet $\Sigma = \{a, b\}$. An ε-NFA is an NFA that can additionally have "silent" transitions labeled with a fresh symbol ε. A word $w \in \Sigma^*$ is *accepted* by an ε-NFA M if there exists a word $w' \in (\Sigma \cup \{\varepsilon\})^*$ such that: 1) w' is accepted by M (when M is considered an NFA over the alphabet $\Sigma \cup \{\varepsilon\}$); and 2) removing all the occurrences of ε in w' gives rise to w. The *language* $L(M) \subseteq \Sigma^*$ of an ε-NFA M is the set of words accepted by M.

An example of an ε-NFA is given below; it is referred to as M_1. Here q_0 is an initial state and q_2 is an accepting state.



Answer the following questions.

(1) Give a regular expression that designates the language $L(M_1)$ of the above ε-NFA M_1.

(2) Give an NFA M_2 over Σ such that: $L(M_2) = L(M_1)$; and M_2 is *ε-free*, that is, there are no transitions labeled with ε in M_2.

(3) Give an ε-NFA M_3 such that

$$L(M_3) = \{w \in \Sigma^* \mid w \text{ contains } aa \text{ as a subword, that is, } w = w_1\, aa\, w_2 \text{ for some } w_1, w_2 \in \Sigma^*\} \ .$$

Your answer may be ε-free.

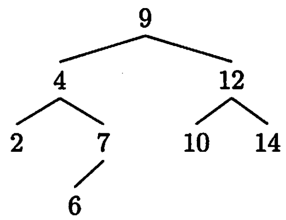
(4) Give an ε-NFA M_4 such that $L(M_4) = L(M_1) \cup L(M_3)$, where M_3 is from Question (3). Your answer may be ε-free.

(5) Give an ε-NFA M_5 such that $L(M_5) = L(M_1) \cap L(M_3)$, where M_3 is from Question (3). Your answer may be ε-free.

Problem 3

Answer the following questions concerning binary search trees. Here, the *height* of a node is defined as the maximum of the graph distance from the node to one of its descendant leaf nodes. For example, a node with no children is of height 0. The *height* of a tree is defined as the height of its root.

(1) Suppose that we have the following binary search tree.



Let us apply the following operations to the above tree, in the shown order.

(i) Insert 3

(ii) Insert 8

(iii) Delete 4

(iv) Delete 9

Depict the state of the tree after each operation.

(2) Answer the minimum and maximum tree heights of a binary search tree with n nodes.

We call a binary search tree *balanced* if every node of it satisfies the following conditions.

- In case the node has two children, the heights of the left and right child subtrees differ by at most 1.
- In case the node has only one child, the height of the child subtree is 0.

Answer the following questions.

(3) Answer the minimum and maximum tree heights of a balanced binary search tree with 7 nodes. Depict a tree with the maximum height, and one with the minimum height.

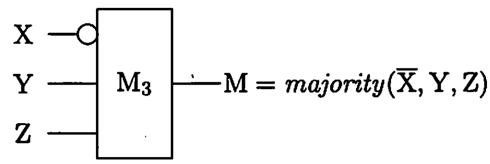
(4) Answer the minimum tree height of a balanced binary search tree with n nodes.

(5) Show that the height of a balanced binary search tree with n nodes is no more than $2\log_2 n$.

Problem 4

The *majority gate* M_3 is a binary logic gate defined as follows: M_3 has 3 inputs and 1 output. The output is 1 if two or three of the inputs are 1; and 0 otherwise.

When you answer circuit designs in the questions below, you can use NOT gates and constants 0 and 1 in addition to M_3 gates, but other gates such as AND or OR cannot be used. You should draw an M_3 gate as a rectangle labeled with M_3, and a NOT gate as a circle, as shown in the following example:



Answer the following questions. Try to make M_3-levels (i.e. the maximum number of serially connected M_3 gates) as small as possible in your answers.

(1) Design and depict the following logic circuits: (a) AND of 2 inputs, (b) OR of 2 inputs, and (c) XOR of 2 inputs.

(2) A 1-bit full adder FA_1 is defined as follows: FA_1 has 3 inputs and 2 outputs called S (sum) and C (carry), respectively. S and C are defined so that $2C + S$ is equal to the sum of the 3 input bits. Design and depict a 1-bit full adder FA_1. Answer its M_3-level, too. You can use the circuits you have designed in Question (1).

(3) Design and depict a 4-bit full adder FA_4. Answer its M_3-level, too.

Here FA_4 takes, as inputs: (i) two unsigned 4-bit integers, and (ii) a carry bit. It outputs the 5-bit sum. You can use the circuits you have designed in Questions (1) and (2).

(4) Design and depict a 4-bit multiplier MUL_4. Answer its M_3-level, too.

Here MUL_4 takes two unsigned 4-bit integers as inputs. It outputs the 8-bit product of the inputs. You can use the circuits you have designed in Questions (1), (2) and (3).

余白 (blank page)

計算などに使ってもよいが，切り離さないこと. Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.