PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS MAY/JUNE 2002

SWANSEA

Computer Science

CS 132 Algorithms and Computation

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS 132
ALGORITHMS AND COMPUTATION

(*Attempt 2 questions out of 3*)

Question 1

(a) Let $A = A[1 \dots N]$ be a sorted array over some linearly ordered domain $(D, <)$, and suppose we want to know whether some specific $d \in D$ occurs as an array entry in A.

(i) Describe and compare the principles of *sequential search* and *binary search* in this setting, including the complexities involved.

(ii) For $N = 10^{10}$, how many steps, roughly, can it take in each case, until d is found (or found not to be in A at all)?

[6 marks]

(b) For the array

$$A = A[1 \dots 9] = (10, 20, 4, 7, 13, 100, 5, 4, 9),$$

over the natural numbers with the usual ordering, sketch the main stages in

(i) InsertionSort(A). (Stages after each completed insertion loop.)

(ii) MergeSort(A). (Splitting pattern and corresponding merges.)

Briefly contrast the basic algorithmic paradigms in InsertionSort and MergeSort.

[8 marks]

(c) (i) Explain in words the main idea of how HeapSort proceeds to sort an array once it has been turned into a heap.

(ii) Sketch the main stages in HeapSort(A) in sorting the array

$$A = (\mathrm{c}, \mathrm{m}, \mathrm{m}, \mathrm{c}, \mathrm{o}, \mathrm{z}, \mathrm{a}),$$

with respect to the alphabetical ordering.
(Display the tree lay-out after each iteration of the main loops
first in Heap(A) and then in the remainder of HeapSort(A).)

[7 marks]

(d) Carefully state the characteristic growth rates for the number of comparisons required in the three sorting procedures encountered above, with a clear statement as to *what* is being measured in its dependency on the array length.

[4 marks]

Question 2

Fix the alphabet $\Sigma = \{a, b, c\}$.

(a) Consider the DFA $\mathcal{A} = (\Sigma, Q, q_0, \delta, A)$: $Q = \{0, 1, 2, 3\}$, $q_0 = 0$, $A = \{0, 2\}$

δ	a	b	c
0	1	0	0
1	3	2	3
2	1	0	0
3	3	3	3

(i) Draw the transition diagram of $\mathcal{A}$.

(ii) Trace the runs of $\mathcal{A}$ on inputs $W_1 = bccabbabac$ and $W_2 = abccbabab$ and determine whether they are accepted or not.

(iii) Describe in words, or by means of a regular expression, the language $L(\mathcal{A})$ that is accepted by $\mathcal{A}$.

(iv) Find an NFA with just two states accepting the same language as $\mathcal{A}$.

[11 marks]

(b) (i) For the regular expression

$$t = \mathbf{a}(\mathbf{b} + \mathbf{c})^*\mathbf{a} + \big((\mathbf{b} + \mathbf{c})(\mathbf{b} + \mathbf{c})\big)^*,$$

precisely describe the language $L(t)$ in plain English, and determine which of the following words are in $L(t)$:

$$W_1 = abcbc \qquad W_3 = ccbbbc$$
$$W_2 = abbba \qquad W_4 = \epsilon$$

[Hint: first look at constituent languages like those denoted by $\mathbf{a}(\mathbf{b} + \mathbf{c})^*\mathbf{a}$ and $\big((\mathbf{b} + \mathbf{c})(\mathbf{b} + \mathbf{c})\big)^*$.]

(ii) Find regular expressions to denote the following Σ-languages:

L_1: words with at least one occurrence of "abc"
L_2: words with an even number of "a" and no "c"

[8 marks]

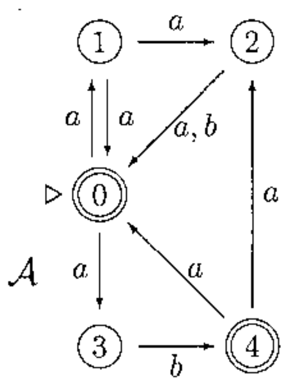
(c) Draw the transition graph of a DFA $\mathcal{A}'$ that accepts precisely those Σ-words with an odd number of "a" and combine the DFA $\mathcal{A}$ from part (a) with $\mathcal{A}'$ to obtain the transition graphs of

(i) a new DFA that accepts the intersection $L(\mathcal{A}') \cap L(\mathcal{A})$.

(ii) a new NFA that accepts the concatenation $L(\mathcal{A}') \cdot L(\mathcal{A})$.

[7 marks]

Question 3



(a) Let $\Sigma = \{a, b\}$ and consider the NFA $\mathcal{A}$ whose transition diagram is given in the figure.

(i) Explicitly write out $\mathcal{A}$ in the usual format $\mathcal{A} = (\Sigma, Q, q_0, \Delta, A)$.

(ii) Consider the input word $W = aaabab$ and display the branching pattern of *all* possible runs of $\mathcal{A}$ on W. Does $\mathcal{A}$ accept W?

(iii) Apply the power set construction to find a DFA $\mathcal{A}^{\text{det}}$ that accepts the same language as $\mathcal{A}$. [Discard all states that are not reachable; you need not draw the transition diagram.]

(iv) Trace the run of $\mathcal{A}^{\text{det}}$ on input $W = aaabab$ and relate it to the runs considered in (ii).

[12 marks]

(b) Discuss how Turing machines crucially differ from finite automata.
Give an example of some simple Σ-language that can be recognised by a Turing machine, but not by a finite automaton, explaining why your sample language is not regular.

[6 marks]

(c) Comment on the impact of non-determinism in polynomially bounded Turing machines and in finite automata. What are the parallels, where do we see fundamental differences?

[7 marks]