Assignment 11

Post Date: 12 Jan 2011   Due Date: 19 Jan 2011, 14:30
You are permitted and encouraged to work in groups of two.

Problem 1: Wildcards

Now, a pattern can contain also wildcards \( \ast \). A wildcard \( \ast \) can stand for arbitrarily many (also zero) characters.

(a) Modify Algorithm Naive-Transition-Function such that it computes a string-matching automaton of a pattern that may contain wildcards.

Consider the pattern \( P = aba \ast bab \) and the input alphabet \( \Sigma = \{a, b, c\} \).

(b) Give the string-matching automaton for the pattern \( P \) computed by your algorithm from (a). Does this automaton find all occurrences in a text of pattern \( P \)?

Problem 2: Transition Function

(a) Give an \( O(m|\Sigma|) \)-time algorithm for computing the transition function \( \delta \) corresponding to a given pattern \( P \) with \( |P| = m \).

(b) Let \( P = abaababa \) be a pattern over the alphabet \( \Sigma = \{a, b\} \). Compute for \( 1 \leq q \leq |P| \) the values \( \pi(q) \), where \( \pi(q) \) is defined as in the lecture.

Problem 3: Cyclic Rotation

Determine in linear time if a text \( T \) is a cyclic rotation of a text \( T' \).

Problem 4: Suffix Tree

(a) Give a naive algorithm for computing a suffix tree for a text \( T \) in \( O(n^2) \), where \( |T| = n \), and where the size of the alphabet is constant.

(b) Use the example \( T = ararat \) to demonstrate your algorithm.

(c) Show that a suffix tree has \( O(n) \) vertices.