—— SOLUTIONS ——

King's College London

UNIVERSITY OF LONDON

This paper is part of an examination of the College counting towards the award of a degree. Examinations are governed by the College Regulations under the authority of the Academic Board.

Enter your candidate number in the box provided above and on the answer book(s) provided. Do this now.

MSc/MSci EXAMINATION

CSMTSP – TEXT SEARCHING AND PROCESSING

MAY 2007

TIME ALLOWED: TWO HOURS.

ANSWER THREE OF THE FIVE QUESTIONS.

NO CREDIT WILL BE GIVEN FOR ATTEMPTING ANY FURTHER QUESTIONS.

ALL QUESTIONS CARRY EQUAL MARKS.

THE USE OF ELECTRONIC CALCULATORS IS **NOT** PERMITTED.

BOOKS, NOTES OR OTHER WRITTEN MATERIAL MAY **NOT** BE BROUGHT INTO THIS EXAMINATION.

NOT TO BE REMOVED FROM THE EXAMINATION HALL

TURN OVER WHEN INSTRUCTED

1. Table of prefixes

Given a word $x = x[0\,.\,.\,m-1]$ the table *Pref* of prefixes of x is defined, for $0 \leq i < m$, by: $\mathit{Pref}[i] = \mathit{lcp}(x, x[i\,.\,.\,m-1])$, which is the maximal length of prefixes common to x and its suffix $x[i\,.\,.\,m-1]$.

a. Give the table *Pref* of the string abaababaabaab.

[10 marks]

Answer

i	0	1	2	3	4	5	6	7	8	9	10	11	12
$x[i]$	a	b	a	a	b	a	b	a	a	b	a	a	b
$Prefix[i]$	13	0	1	3	0	6	0	1	5	0	1	2	0

[unseen]

b. How do you characterize the positions of occurrences of x in another string y using the table *Pref* of the string $x\$y$ (assuming that the symbol \$ does not occur in x nor in y)?

[5 marks]

Answer

The string x occurs at position i on y iff $\mathit{Pref}[i + |x| + 1] = |x|$. [unseen]

c. Let i be a position on x, $0 \leq i < m$. Let $j = i + \mathit{Pref}[i]$. What can you say about the border of $x[0\,.\,.\,j-1]$?

[5 marks]

Answer

The word $x[i\,.\,.\,j-1]$ is a prefix of x, then a border of $x[0\,.\,.\,j-1]$, but not always the longest as shows the next example.

For $x = abaaba$ and $i = 5$, $5 + \mathit{Pref}[5] = 5 + 1 = 6$, $x[5\,.\,.\,5] =$ a is a border of x, but its (longest) border is aba. [unseen]

d. A square is a word of the form uu where u is a non-empty word. Indicate how to find all squares that are prefixes of x using its table *Pref*.

[5 marks]

Answer

$x[0\,.\,.\,2i-1]$ is a square iff $\mathit{Pref}[i] \geq i$. [unseen]

e. Let *Pref* be the table of prefixes of x. Let f, g, i be three positions on x for which $g = f + Pref[f]$ and $f < i < g$. What is the value of $Pref[i]$ when $Pref[i - f] \neq g - i$?

[10 marks]

Answer

If $Pref[i-f] < g-i$, $Pref[i] = Pref[i-f]$. If $Pref[i-f] > g-i$, $Pref[i] = g-i$. [unseen]

f. Write in your own words a linear-time algorithm to compute the table *Pref* of x.

[15 marks]

Answer

```
Prefixes(x,m)
    Pref[0] = m
    g = 0
    for i = 1 to m − 1 do
        if(i < g and Pref[i − f] ≠ g − i)
            Pref[i] = min{Pref[i − f], g − i}
        else
            (f, g) = (i, max{g, i})
            while(g < m and x[g] = x[g − f])
                g = g + 1
            Pref[i] = g − f
    return Pref
```

[unseen, 10 marks]
The algorithm runs in linear time since positive letter comparisons increase the value of g that never decreases and goes from 0 to m, and negative comparisons leads to incrementing i whose values go from 0 to $m-1$. [unseen, 5 marks]

— SOLUTIONS —

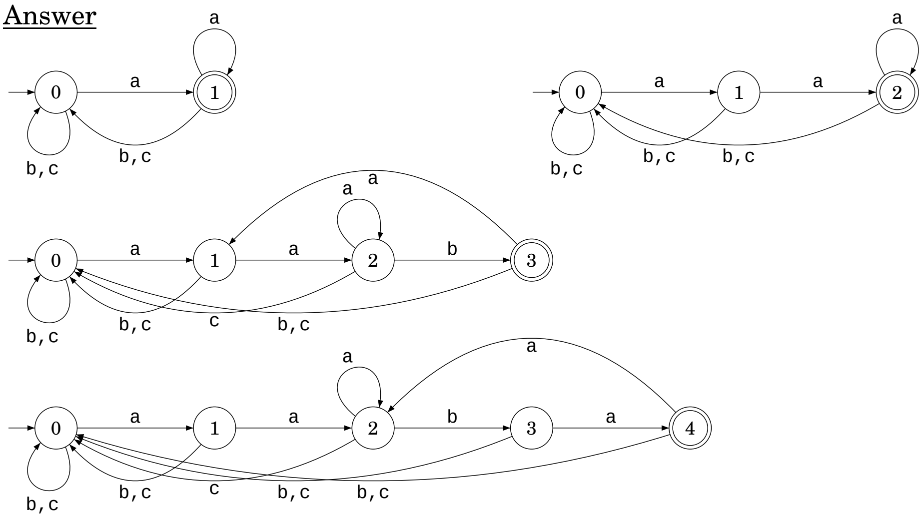
2. String Matching Automaton

We consider the alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. For a string $x \in \Sigma^*$, the string matching automaton of x, $SMA(x)$, is the minimal deterministic automaton accepting the language $\Sigma^* x$. Its initial state is denoted by $initial$, its terminal state by $terminal$, and its transition function by δ.

a. Design the string matching automata $SMA(\mathsf{a})$, $SMA(\mathsf{aa})$, $SMA(\mathsf{aab})$, $SMA(\mathsf{aaba})$.

[10 marks]

Answer



[in lectures]

b. Describe how to build efficiently the automaton $SMA(x\sigma)$ from the automaton $SMA(x)$ when $x \in \Sigma^*$ and $\sigma \in \Sigma$.

[15 marks]

Answer

Let $r = \delta(terminal, \sigma)$. The automaton is transformed by adding a new state s and keeping the same transitions except that $\delta(terminal, \sigma)$ is set to s. Then, the transitions from s reproduce those from r, that is: $\delta(s, \tau) = \delta(r, \tau)$ for every $\tau \in \Sigma$. Finally, s becomes the only terminal state. [in lectures]

c. List all the forward arcs of $SMA(\mathsf{aaba})$. List all its backward arcs.

[10 marks]

Answer

Its forward arcs are: $(0, \mathsf{a}, 1)$, $(1, \mathsf{a}, 2)$, $(2, \mathsf{b}, 3)$, $(3, \mathsf{a}, 4)$.

Its backward arcs are: $(2, \mathsf{a}, 2)$, $(4, \mathsf{a}, 2)$. [unseen]

d. Give a formal characterization of a backward arc of $SMA(x)$. Prove that for any string x the automaton $SMA(x)$ has no more than $|x|$ backward arcs. For each positive integer m give a string of length m whose string matching automaton has exactly m backward arcs.

[15 marks]

Answer

A backward arc of $SMA(x)$ is of the form $(u, \tau, v\tau)$ for some strings u, v prefixes of x, where v is proper suffix of u. Therefore v is a border of u. Let $p(u, \tau) = |u| - |v|$ be the corresponding period of u. [in lectures, 5 marks]

We show that two different backward arcs are associated with two different periods. Let $(u, \tau, v\tau)$ and $(u', \tau', v'\tau')$ be two backward arcs such that $p(u, \tau) = p(u', \tau')$.

If $v = v'$ then $|u| = |u'|$ and $u = u'$. We have also $\tau = x[|v|] = x[|v'|] = \tau'$. Thus the backward arcs coincide.

If $v \neq v'$, let us consider for example that v' is a proper prefix of v. By definition of the backward arc $(u', \tau', v'\tau')$ we have $\tau' = x[|v'|] \neq x[|u'|]$. But since $p(u, \tau)$ is a period of u we have $\tau' = x[|v'|] = x[|v'| + p(u, \tau)] = x[|v'| + p(u', \tau')] = x[|u'|]$ a contradiction.

Consequently p is injective, and since it has at most $|x|$ possible values, there are at most $|x|$ possible backward arcs. [in lectures, 5 marks]

For each positive integer m, the automaton $SMA(\mathtt{ab}^{m-1})$ has exactly m backward arcs. [in lectures, 5 marks]

3. Linear-time suffix sorting

Let y be a string of length n.

a. List the nonempty suffixes of the string ababababaa in lexicographic order assuming a $<$ b.

[5 marks]

Answer

a,aa,abaa,ababaa, abababaa,baa,babaa,bababaa. [unseen]

b. Let P_{01} be the positions on y of the form $3q$ or $3q+1$. Let P_2 be the positions on y of the form $3q+2$. Describe the four steps of the Skew algorithm to sort the suffixes of y.

[20 marks]

Answer

1. Sort the position in P_{01} according to their associated 3-grams. Let $t[i]$ be the rank of i in the sorted list.
2. Recursively sort the suffixes of $t[0]t[3]\ldots t[1]t[4]\ldots$. For a position i in P_{01}, let $s[i]$ be the rank of its associated suffix in the sorted list of them, L_{01}.
3. Sort the positions j in P_2. Let L_2 be the sorted list.
4. Merge lists L_{01} and L_2.

[in lectures, 5 marks for each step]

c. Let L_{01} be the list of positions of P_{01} sorted according to their associated suffixes; let $s[i]$ be the rank of i in L_{01}. Describe how to sort P_2 in time $O(\mathrm{card}P_2)$.

[10 marks]

Answer

Sorting elements j of P_2 remains to sort their associated pairs $(y[j], s[j+1])$. This can be done in linear time using radix sort. [in lectures]

d. In addition to L_{01} and s in Question 3.c, let L_2 be the list of positions of P_2 sorted according to their associated suffixes. Describe how to compare i in L_{01} with j in L_2 in constant time. How long does it take?

[15 marks]

Answer

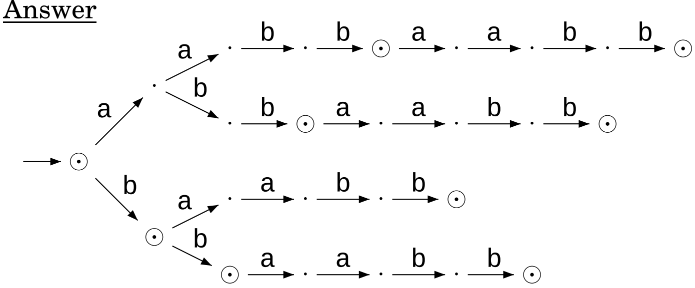
If i is of the form $3q$, $i+1$ and $j+1$ are in L_{01}, thus $s[i+1]$ and $s[j+1]$ are defined. Comparing i and j amounts to compare $(y[i], s[i+1])$ and $(y[j], s[j+1])$.
If i is of the form $3q+1$, $i+2$ and $j+2$ are in L_{01}, thus $s[i+2]$ and $s[j+2]$ are defined. Comparing i and j amounts to compare $(y[i]y[i+1], s[i+2])$ and $(y[j]y[j+1], s[j+2])$. [in lectures, 10 marks]
In both cases comparisons are done in constant time. [in lectures, 5 marks]

4. Suffix trie and suffix tree

a. Design the trie of suffixes of the word $y = \texttt{aabbaabb}$.

[10 marks]

Answer



[unseen]

b. Give an example of a word of length n on the alphabet $\{\texttt{a}, \texttt{b}\}$ having a trie of suffixes of size $\Omega(n^2)$.

[10 marks]

Answer

The trie of the word $a^{n/4}b^{n/4}a^{n/4}b^{n/4}$, for two distinct letters a and b, has at least $n/4$ branches each of them having $n/4$ nodes. Which gives $(n/4)^2 = \Omega(n^2)$ nodes. [in lectures]

c. Design an algorithm to compact the trie of suffixes of a word into its suffix tree.

[20 marks]

Answer

The following procedure compacts a trie T, even if suffix links are defined on states.

Compact(trie T)
    $r \leftarrow$ root of T
    for each arc (r, a, p) do
        Compact(subtrie of T rooted at p)
        if(p has exactly one child)
            $q \leftarrow$ that child
            $u \leftarrow$ label of (p, q)
            replace p by q as child of r
            set $a \cdot u$ as label of (r, q)

[unseen, 10 marks for correct tree traversal, 10 marks for correct node deletion]

d. Describe possible data structures required to implement the suffix tree of a word y.

[10 marks]

Answer

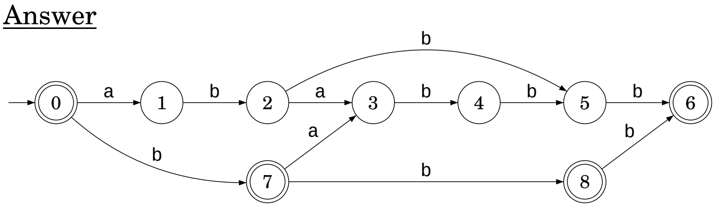
Each node or state p of the tree can be implemented as a structure containing two pointers: the first pointer to implement the suffix link; the second pointer to give access to the list of arcs outgoing state p. The list of arcs can contain 4-tuples in the form (a, i, ℓ, q) where a is a letter, i and ℓ are integers, and q is a pointer to a state. They are such that (p, u, q) is an arc of the automaton with $a = y[i]$ and $u = y[i \mathinner{.\,.} i + \ell - 1]$. [unseen]

5. Suffix automaton

a. Design $SA($`ababbb`$)$, the suffix automaton of the string `ababbb`.

[10 marks]

Answer

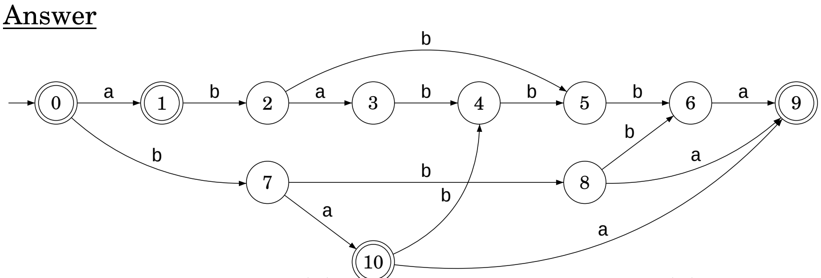


[unseen]

b. Indicate how to modify the automaton of question 5.a to get $SA($`ababbba`$)$.

[10 marks]

Answer



[unseen]

c. Let p be a state of $SA(y)$, for a string y. Let $SA_p(y)$ be the automaton obtained from $SA(y)$ by considering p as the only initial state. How do you characterize the words accepted by the automaton $SA_p(y)$?

[10 marks]

Answer

Words accepted by $SA_p(y)$ are suffixes of y that start with any of the words labelling paths from the initial state to p. [unseen]

d. Let p be a state of $SA(y)$ and let $SA_p(y)$ be as in question 5.c. Let $X(p)$ be the number of words accepted by $SA_p(y)$ considering that all its states are terminal states. Give a recurrence relation to compute $X(p)$ from the $X(q)$s where the qs are targets of transitions from p.

[10 marks]

Answer

$$X[p] = \begin{cases} 1 & \text{if } deg(p) = 0, \\ 1 + \sum_{(p,v,q)\in F}(|v| - 1 + X[q]) & \text{otherwise,} \end{cases}$$

where F is the set of arcs of the automaton. [unseen]

e. What is the running time of an algorithm using the recurrence of question 5.d to compute the number of strings accepted by $SA(y)$? Explain why.

[10 marks]

<u>Answer</u>

The computation is done during a traversal of the automaton starting in the initial state. Since no transition is executed, if the implementation is by lists of successors, the running time is $O(|y|)$. It is $O(\#A \times n)$ if a transition matrix is used. [unseen]