—— SOLUTIONS ——

King's College London

UNIVERSITY OF LONDON

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

MSc EXAMINATION

CSMTSP – TEXT SEARCHING AND PROCESSING

MAY 2003

TIME ALLOWED: TWO HOURS.

ANSWER ALL THREE QUESTIONS

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. Borders of strings

a. Report all periods and borders of the string abaababaabaababaaba.

[10 marks]

Answer

Periods are 8, 13, 16, 18, 19; corresponding borders are words abaababaaba, abaaba, aba, a, ε.

b. Given a string y, design an algorithm that computes the table $Border$. Recall that, for $y = y[1..n]$ and $1 \leq i \leq n$, $Border[i]$ is the maximal length of (proper) borders of $y[1..i]$.

[15 marks]

Answer

```
ComputeBorders(y : string, m : integer)
Border[0] ← −1
for i ← 1 to m do
    j ← Border[i − 1]
    while j ≥ 0 and y[i] ≠ y[j + 1] do
        j ← Border[j]
    Border[i] ← j + 1
```

c. Give the output of your algorithm in question 1.b for the input: abaababaabaababaaba.

[5 marks]

Answer

ϵ	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b	a
-1	0	0	1	1	2	3	2	3	4	5	6	4	5	6	7	8	9	10	11

d. Given the string y, let $Pref$ be the table defined as: $Pref[i]$ is the maximal length of prefixes common to y and its suffix starting at position i. Give the table $Pref$ for the input of question 1.c, and an expression for $Border[j]$ using $Pref$.

[10 marks]

Answer

a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b	a
19	0	1	3	0	6	0	1	11	0	1	3	0	6	0	1	3	0	1

Let $I = \{i \mid 0 \leq i \leq j \text{ and } i + Pref[i] - 1 \geq j\}$, then $Border[j] = \begin{cases} 0 & \text{if } I = \emptyset, \\ j - \min I + 1 & \text{otherwise.} \end{cases}$

e. The overlap between strings y and x is the maximal length of strings u that are both suffixes of y and prefixes of x. Describe how to compute the overlap between y and x in time $O(|y| + |x|)$.

[10 marks]

Answer

Apply MP algorithm with pattern x and text the suffix of y of length at most $|x|$. When the algorithm stops the pointer on x give the answer.

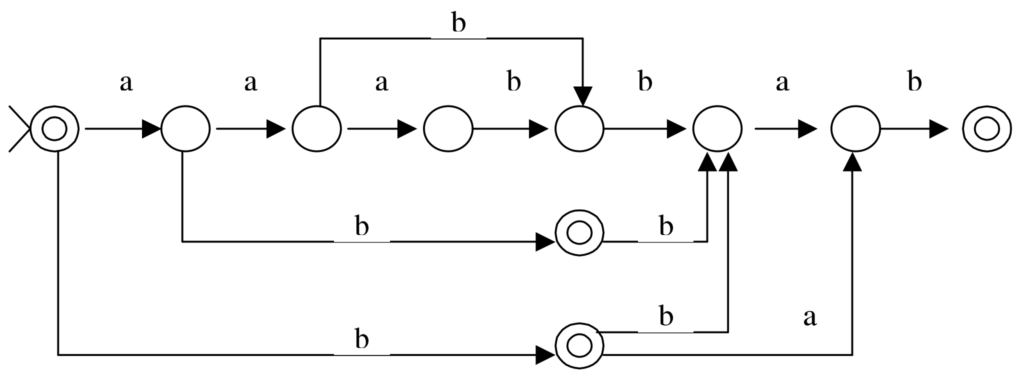
—— SOLUTIONS ——

2. Suffix automaton

a. Design $SA(\texttt{aaabbab})$, the suffix automaton of the string aaabbab.

[10 marks]

Answer

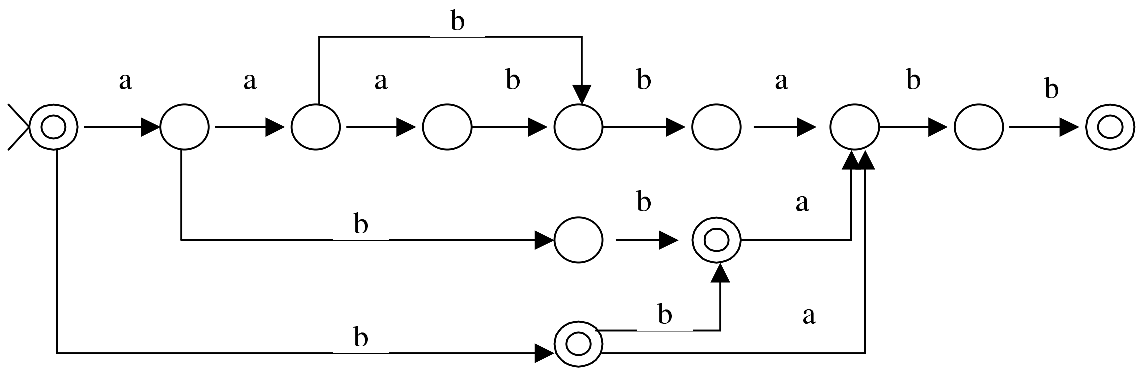


b. Indicate how the automaton of question 2.a is modified to get $SA(\texttt{aaabbabb})$.

[20 marks]

Answer

Suffix link $f(aaabbab)$ =state(ab), and arc (ab, b) is non-solid. Therefore, this arc is redirected onto a cloned state from $(aaabb)$. Similarly, the arc (b, b) is redirected to the same state.



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 strings accepted by the automaton $SA_p(y)$?

[10 marks]

Answer

Strings accepted by $SA_p(y)$ are suffixes of y that start with any of the strings labeling paths from the initial state to p.

d. Let p be a state of $SA(y)$ and let $SA_p(y)$ be as in question 2.c. Consider that all states of $SA(y)$ (and then $SA_p(y)$) are terminal. Let $X(p)$ be the number of strings accepted by $SA_p(y)$. Give a recurrence relation to compute $X(p)$ from the $X(q)$'s where q's are targets of transitions from p.

[5 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.

e. What is the complexity of an algorithm using the recurrence of question 2.d to compute the number of strings accepted by $SA(y)$? Explain your answer.

[5 marks]

Answer

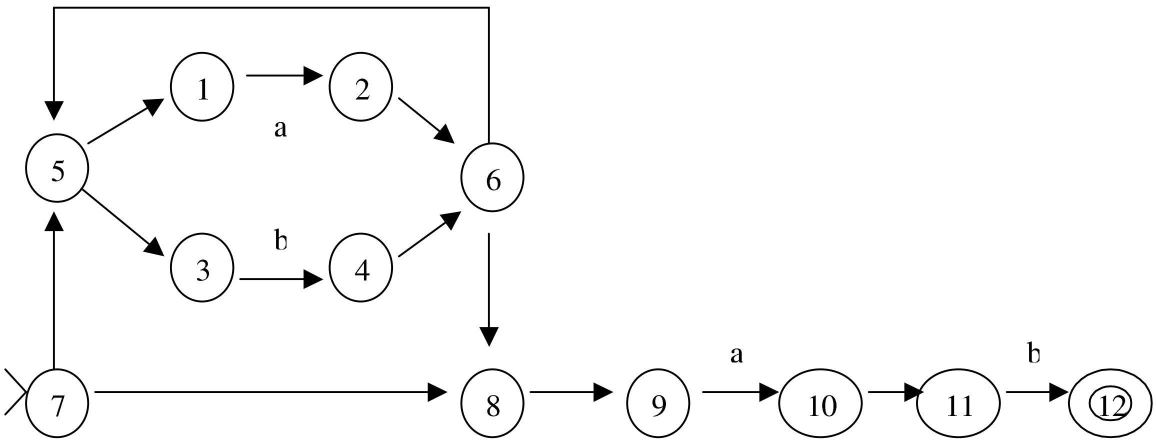
The computation is done during a traversal of the automaton starting in state p. Since no transition is executed, if the implementation is by lists of successors, the running is $O(|y|)$. It is $O(\#A \times n)$ if a transition matrix is used.

3. Regular matching

a. Design the non-deterministic automaton $\mathcal{A}$ associated with the regular expression $(\mathtt{a}+\mathtt{b})^*\mathtt{ab}$ which is obtained by Thompson's construction.

[10 marks]

Answer



b. Describe data structures to efficiently implement the automata obtained by Thompson's construction.

[10 marks]

Answer

Use an array T indexed by states $T[p] = (a, q)$ or (ϵ, q, r), where a is a symbol, q and r are targets of arcs from p.

c. Simulate the regular pattern matching algorithm using the automaton $\mathcal{A}$ of question 3.a on the string `aabbab`.

[10 marks]

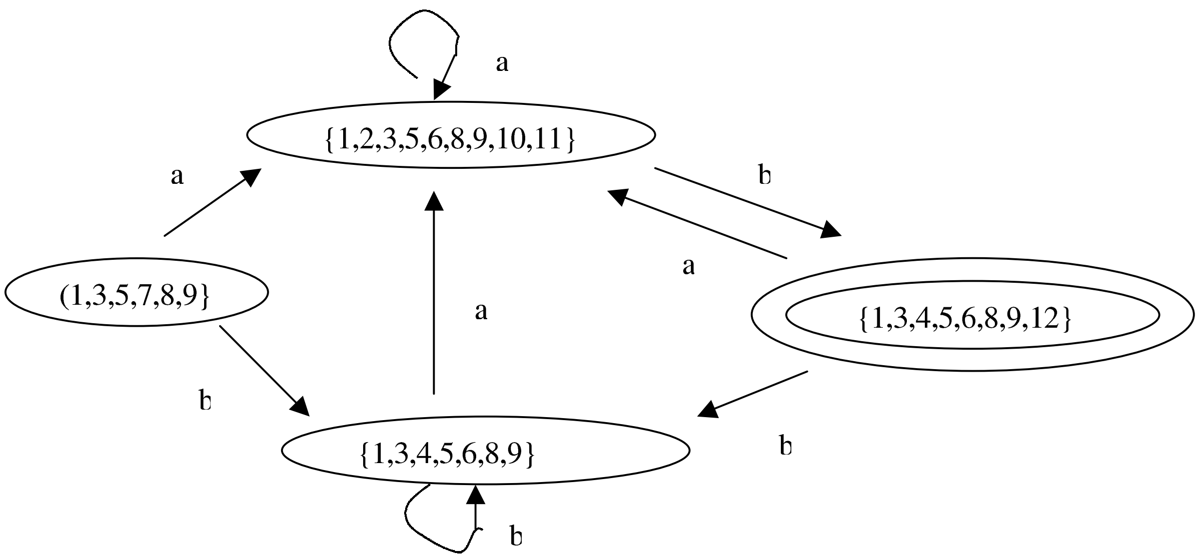
Answer

closure$(7) = 1, 3, 5, 7, 8, 9$
transition by a: $\{2, 10\}$
closure: $\{1, 2, 3, 5, 6, 8, 9, 10, 11\}$
transition by a: $\{2, 10\}$
closure: $\{1, 2, 3, 5, 6, 8, 9, 10, 11\}$
transition by b: $\{4, 12\}$
closure: $\{1, 3, 4, 5, 6, 8, 9, 12\}$
transition by b: $\{4\}$
closure: $\{1, 3, 4, 5, 6, 8, 9\}$
transition by a: $\{2, 10\}$
closure: $\{1, 2, 3, 5, 6, 8, 9, 10, 11\}$
transition by b: $\{4, 12\}$
closure: $\{1, 3, 4, 5, 6, 8, 9, 12\}$

d. Design the deterministic automaton equivalent to the non-deterministic automaton $\mathcal{A}$ of question 3.a. Use the subset construction.

[10 marks]

Answer



e. State the complexity (time and space) of matching regular patterns with Thompson's automata and their deterministic versions.

[10 marks]

Answer

Searching for a regular expression of size r in a text of length n.
Thompson's: time $O(rn)$, space $O(r)$
Deterministic automaton: time $O(n)$, space $O(2^r)$