—— SOLUTIONS ——

King's College 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.

BSc EXAMINATION

6CCS3TSP – TEXT SEARCHING AND PROCESSING

MAY 2010

TIME ALLOWED: TWO HOURS.

ANSWER **TWO** OF THE **THREE** 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

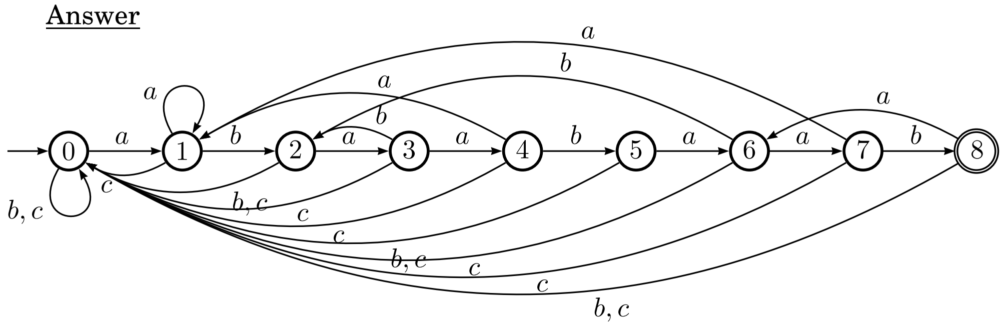
1. Matching Automata

We consider the alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. For a string $x \in \Sigma^*$, the string matching automaton of x, $\mathrm{SMA}(x)$, 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. Draw the string matching automaton of the string abaabaab.

[10 marks]

Answer



[unseen]

b. Describe how to build efficiently the automaton $\mathrm{SMA}(xa)$ from the automaton $\mathrm{SMA}(x)$ when $x \in \Sigma^*$ and $a \in \Sigma$.

[15 marks]

Answer

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

c. List all the forward arcs of $\mathrm{SMA}(\mathsf{abaabaab})$. List all its backward arcs. What is the maximal number of backward arcs in the string matching automaton of a string of length n?

[10 marks]

Answer

Forward arcs: $(0, \mathsf{a}, 1)$, $(1, \mathsf{b}, 2)$, $(2, \mathsf{a}, 3)$, $(3, \mathsf{a}, 4)$, $(4, \mathsf{b}, 5)$, $(5, \mathsf{a}, 6)$, $(6, \mathsf{a}, 7)$, $(7, \mathsf{b}, 8)$.

Backward arcs: $(1, \mathsf{a}, 1)$, $(3, \mathsf{b}, 2)$, $(4, \mathsf{a}, 1)$, $(6, \mathsf{b}, 2)$, $(7, \mathsf{a}, 1)$, $(8, \mathsf{a}, 6)$. [unseen]

The maximal number of backward arcs is n, reached for example for the string ab^{n-1}. [in lectures]

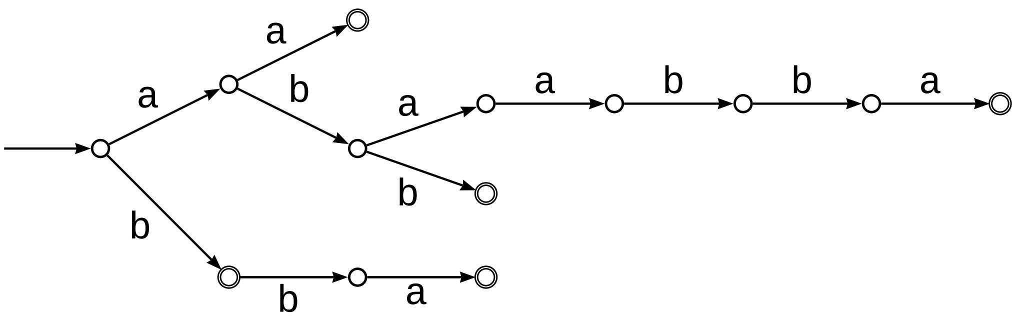
d. Draw the trie of the set {aa, abaabba, abb, bba}. Mark its terminal states.

Draw the implementation with failure links of the dictionary-matching automaton DMA({aa, abaabba, abb, b, bba}). Mark its terminal states.

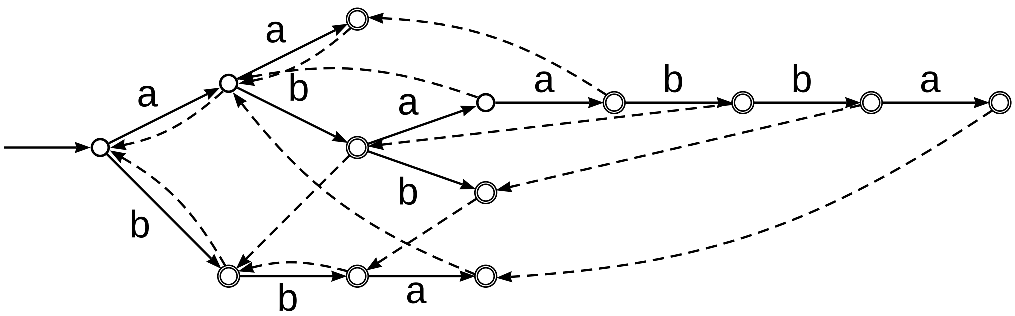
Define the notion of a failure link $f(p)$ on a state (node) p of the trie of a finite set X of strings.

[15 marks]

<u>Answer</u>



[unseen]



[unseen]

Let p be a state of the trie of X distinct from the root. Let $u \in \Sigma^+$, be the label of the path from the root to state p. Then the failure state $f(p)$ of state p is the state of the trie whose path from the root is labelled by the longest possible proper suffix of u. [in lectures]

2. Doubling

Let y be a fixed text of length n.

For a word u and a positive integer k, $First_k(u)$ is u if $|u| \leq k$ and is $u[0\,..\,k-1]$ otherwise. The integer $R_k[i]$ is the rank of $First_k(y[i\,..\,n-1])$ inside the sorted list of all $First_k(u)$ where u is a nonempty suffix of y (ranks are numbered from 0).

a. Give R_1, R_2, R_3, R_4, R_8 for the word aababbabba, assuming a < b.

[10 marks]

Answer

i	0	1	2	3	4	5	6	7	8	9
$y[i]$	a	a	b	a	b	b	a	b	b	a
$R_1[i]$	0	0	1	0	1	1	0	1	1	0
$R_2[i]$	1	2	3	2	4	3	2	4	3	0
$R_3[i]$	1	2	5	3	6	5	3	6	4	0
$R_4[i]$	1	2	5	3	7	5	3	6	4	0
$R_8[i]$	1	2	7	4	9	6	3	8	5	0

[unseen]

b. State the doubling lemma and prove it.

[15 marks]

Answer

Lemma 1 *$Rank_{2k}[i]$ is the rank of the pair $(Rank_k[i], Rank_k[i+k])$ in the sorted list of these pairs.*

[bookwork]

Proof. Let i be a position on y and let $u = First_{2k}(y[i\,..\,n-1])$. Let j be a position on y and let $v = First_{2k}(y[j\,..\,n-1])$. We show that $u \leq v$, which is equivalent to $Rank_{2k}[i] \leq Rank_{2k}[j]$, iff $(Rank_k[i], Rank_k[i+k]) \leq (Rank_k[j], Rank_k[j+k])$.

First case: $First_k(u) < First_k(v)$. This is equivalent to $Rank_k[i] < Rank_k[j]$ so the result holds in this case.

Second case: $First_k(u) = First_k(v)$. This is equivalent to $Rank_k[i] = Rank_k[j]$. Then the comparison between u and v depends only on the second halves of these words; in other terms, $Rank_{2k}[i] \leq Rank_{2k}[j]$ is equivalent to $Rank_k[i+k] \leq Rank_k[j+k]$. [unseen]

c. Describe an efficient algorithm to compute R_{2k} from R_k. What is its running time?

[15 marks]

Answer

Two steps: first sort positions i according to the pairs $(R_k[i], R_k[i+k])$; then assign the same R_{2k} rank to positions associated with the same pair.

First step can be implemented by bucket sort (count sort) in linear time; second step is obvious and runs also in linear time. [bookwork]

d. Define the two arrays SUF and LCP composing the Suffix Array of the string y. Using the result of Question 2.c, give the running time of the induced algorithm to compute the array SUF. Justify your answer.

[10 marks]

Answer

The array SUF contains the permutation of suffix positions in increasing order of the suffixes:

$$y[\mathrm{SUF}[0]\,..\,n-1] < y[\mathrm{SUF}[1]\,..\,n-1] < \ldots < y[\mathrm{SUF}[n-1]\,..\,n-1]$$

and the LCP array is defined by:

$$\mathrm{LCP}[i] = |\mathrm{lcp}(y[\mathrm{SUF}[i-1]\,..\,n-1], y[\mathrm{SUF}[i]\,..\,n-1])|$$

where $\mathrm{lcp}(u, v)$ is the longest common prefix of u and v.

The runtime of the induced algorithm is $O(n \times \log n)$ because there are $\lceil \log n \rceil$ steps and each step can be implemented to run in $O(n)$ from answer to Question 2.c. [bookwork]

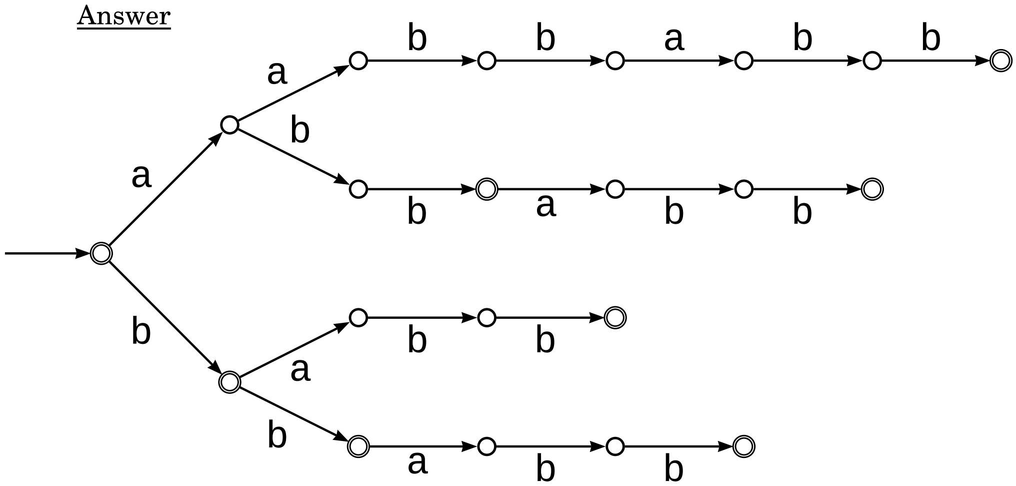
3. Suffix trie and suffix tree

a. Design the trie of suffixes of the string $y = \texttt{aabbabb}$.

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

[10 marks]

Answer



[unseen]

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]

b. Define the notion of Suffix Tree of a string y. Define the notion of Suffix Link for the nodes of the tree.

[10 marks]

Answer

The Suffix Tree of a string y is the compacted version of its Suffix Trie. It has the following characteristics, which make the tree unique for a given string:

- it is an automaton whose initial state is the root and arcs are labelled by nonempty factors of y,
- each terminal node is associated with a suffix of y, label of the path from the root to it,
- no other string labels such a path,
- internal nodes either have two children/successors or have only one child/successor and are terminal,
- when two arcs starts from the same node their labels starts by two different letters.

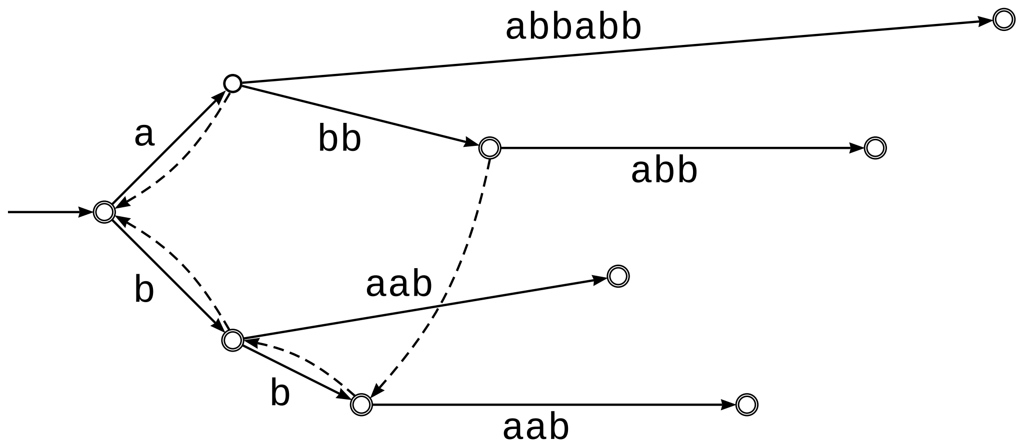
[unseen]

If a node p is associated with a nonempty factor au of y (a letter, u string), its suffix target $s(p)$ is associated with the factor u. The Suffix Link is the function s defined on internal nodes of the tree, except the root.

c. Draw the Suffix Tree of the string $y = \mathsf{aabbabb}$ with the Suffix Link.

[10 marks]

<u>Answer</u>



[unseen]

d. Describe a possible data structure for implementing the suffix tree of a word y.

[10 marks]

<u>Answer</u>

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]

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

[10 marks]

<u>Answer</u>

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

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

[unseen]