Petr Lisoněk

On related transducers

Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica, Vol. 29 (1990), No. 1, 291,292--293,294--299

Persistent URL: http://dml.cz/dmlcz/120238

ACTA UNIVERSITATIS PALACKIANAE OLOMUCENSIS
FACULTAS RERUM NATURALIUM
1990 MATHEMATICA XXIX VOL. 97

Katedra výpočetní techniky
přírodovědecké fakulty Univerzity Palackého v Olomouci
Vedoucí katedry: Doc.RNDr.František Kolíba, CSc.

ON RELATED TRANSDUCERS

PETR LISONĚK
(Received April 30, 1989)

1. INTRODUCTION

The general string-matching problem, searching for a pattern in a string, has been widely studied since the early seventies. Two fundamentally different approaches have been used in solving it according to whether it is the pattern or the string which changes more often.

Firstly, consider the fixed pattern w (over an alphabet A) we are searching for. It is easy to build the deterministic finite automaton (DFA) M which recognizes the language A^*w - the set of all words over A ending with w . Then we can let M work on the string inside which the pattern w is expected and get a "real-time" search algorithm (see, e.g., [2], [3]).

The subject of this paper is the other approach supposing that we are searching for a varying pattern in a fixed string, say x. In this case efficiency is reached by preprocessing x.

The set F(x) of all subwords (factors) of x, being finite, is recognized by some minimal DFA M(x). Furthermore, in most of applications we are interested not only in the basic information (if the pattern z is a factor of x or not) but a position of z in x is needed, too. The function pos(z,x) taking as its value the position of first occurrence of z in x (undefined otherwise) is left sequential and can be computed by a (sequential) transducer having M(x) as its underlying automaton. Such transducers have been examined in detail by Crochemore in [1].

Basic concepts and results from there will be briefly presented in Section 2. However, the careful reader is asked to get familiar with the complete article [1]. This is especially needed for understanding the proof of our Theorem (Section 3).

2. FACTOR TRANSDUCERS

All the words considered in this and the latter sections are elements of the free monoid A^* generated by some finite alphabet A. The empty word of A^* is denoted by 1. Letters of A will be denoted by a,b,c,... and words from A^* by x,y,z,u, v,w,... . The notation |w| is used for the length of the word w.

The set of all factors (substrings) of x is

$$F(x) = \{ w \in A^* \mid \exists\, u,v \in A^* \ , \ x = uwv \} \ .$$

The left sequential function

$$p_x(y) = \text{shortest } w \in A^* \text{ such that } \exists\, u,v \in A^* \ , \quad w = uy \quad \text{and} \quad x = wv$$

is defined from F(x) to the set of all prefixes of x.

In Section 1 remembered left-sequential position function is now defined from F(x) to N as

$$pos(z,x) = |p_x(z)| - |z| \ .$$

The set F(x), being finite, is recognized by the minimal DFA M(x). The transducer having M(x) as its underlying automaton and associated with the function pos(z,x) is called (minimal) factor transducer of the word x and will be denoted as C(x). An example of the factor transducer is shown in the following figure.

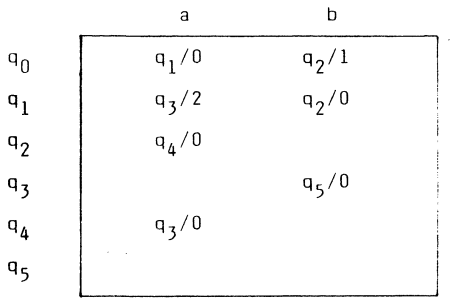

Fig.1. Factor transducer C(abaab)

For example,

pos(aab,abaab) = 0+2+0 = 2 .

For $x \in A^*$, the number of states of C(x) is denoted by e(x). E.g., e(abaab) = 6. But e(baaba) = 7 (construct C(baaba) !).

In [1], Crochemore stated two important propositions about factor transducers:

Proposition 1. The number e(x) satisfies

if $|x| \leq 3$, $e(x) = |x|+1$

if $|x| > 3$, $|x|+1 \leq e(x) \leq 2|x|-2$ and

$e(x) = 2|x|-2$ iff $x \in ab^*c$, $a \neq b$, $b \neq c$.

(see [1], p.73)

Proposition 2. On a given alphabet A, factor transducer C(x) can be built in time and space both linear in the length of x.
(see [1], p.78)

Moreover, for each $x \in A^*$, $a \in A$ one can compute $e(xa)$ from $e(x)$ and some additional facts about $F(x)$ and $C(x)$ ([1], p.72).

3. RELATED TRANSDUCERS

Given factor transducer (let's abbreviate FT) $C(w)$ of a word w, consider FT $C(w^R)$ of the mirror image of w. We shall call the FT's $C(w)$ and $C(w^R)$ to be related. Since related transducers can be used to similar purposes (see Section 4), our main goal in this paper is to compare their sizes, $e(x)$ and $e(x^R)$, for various $x \in A^*$. Returning to example in Section 2, we can find out a little difference between $e(w)$ and $e(w^R)$ for $w = abaab$.

Now we ask, how large this difference can be for other words from A^*. It is a bit surprising that there are words w such that the two values $e(w)$ and $e(w^R)$ are "spanned" over nearly the whole interval $\langle |w|+1 , 2|w|-2 \rangle$ determined for them in Proposition 1.

Theorem. Let $w = a^2b^n ab^{n+1}$ for some positive integer n .
Then $e(w) = 2|w|-4$,
$e(w^R) = |w^R|+2 = |w|+2$.

Proof.

1) Firstly consider $w = a^2b^n ab^{n+1}$.

It follows from Proposition 1. that $e(a^2b) = 4$. Since $s(a^2b) = 1$ (the empty word) and $b \in F(a^2b)$, $e(a^2b^2) = e(a^2b) + 1 = 5$. Now set $x = a^2b^i$. Then $s(x) = b^{i-1}$, $b^{i-1}b \in F(x)$ and thus $e(a^2b^{i+1}) = e(xb) = e(x) + 1$. Therefore $e(a^2b^n) = n+3$ for every n. (Cf. with the automaton $M(a^2b^n)$.)

Now let $x = a^2b^n$. Then $s(x) = b^{n-1}$, $safe(x) = 1$. Since $b^{n-1}a \notin F(x)$, $e(xa) = e(x) + |b^{n-1}| + 1 = (n+3) + (n-1) + 1 = 2n + 3$.

For $x = a^2b^n a$ we have $s(x) = a$, $ab \in F(x)$ which gives $e(a^2b^n ab) = e(xb) = e(x) + 1 = 2n + 4$. In general, if $x = a^2b^n ab^i$ $(i \leq n-1)$ then $s(x) = b^i$, $b^i b \in F(x)$ and so

$e(a^2b^nab^{i+1}) = e(xb) = e(x) + 1 = 2n + i + 4$. Thus $e(a^2b^nab^n) = 3n + 3$. (Cf. with the DFA $M(a^2b^nab^n)$.)

For $x = a^2b^nab^n$ we have $s(x) = ab^n$, $safe(x) = a$ and since $ab^nb \notin F(x)$, $e(a^2b^nab^{n+1}) = e(xb) = e(x) + |a^{-1}ab^n| + 1 =$ $= 3n + 3 + n + 1 = 4n + 4$.

The proof of

$$e(w) = 4n + 4 = 2|w| - 4 \qquad \text{for } w = a^2b^nab^{n+1}$$

is now completed.

2) Now examine the reversal of w, $w^R = b^{n+1}ab^na^2$.

It is a bit simpler to prove that $e(w^R) = |w^R| + 2$:

It is easy to see that $e(b^{n+1}) = n + 2$. For this word we have $s(b^{n+1}) = b^n$ and $safe(b^{n+1}) = b^n$. Since $b^na \notin F(b^{n+1})$, $e(b^{n+1}a) = e(b^{n+1}) + |(b^n)^{-1}b^n| + 1 = n + 3$. Furthermore, if $x = b^{n+1}a$, $s(x) = 1$ and $e(b^{n+1}ab) = e(xb) = e(x) + 1 = n + 4$. In general, for $x = b^{n+1}ab^i$ (assuming $i \leq n-1$) we obtain $s(x) = b^i$, $b^ib \in F(x)$ and $e(b^{n+1}ab^{i+1}) = e(xb) = e(x) + 1$ from where we can derive $e(b^{n+1}ab^n) = 2n + 3$.

Let's take now this word, $x = b^{n+1}ab^n$: $s(x) = b^n$, $b^na \in F(x)$. Hence $e(b^{n+1}ab^na) = e(xa) = e(x) + 1 = 2n + 4$.

Finally, examine $x = b^{n+1}ab^na$: From $C(b^{n+1}ab^na)$ we see that $s(x) = b^na$, $safe(x) = b^n$ and so (remember $b^na^2 \notin F(x)$) we compute $e(w^R) = e(b^{n+1}ab^na^2) = e(xa) = e(x) +$ $+ |(b^n)^{-1}b^na| + 1 = 2n + 4 + 1 + 1 = 2n + 6$.

This completes part 2) of our proof since for $w^R =$ $= b^{n+1}ab^na^2$ we have obtained

$$e(w^R) = 2n + 6 = |w^R| + 2 = |w| + 2 \quad .$$

4. CONCLUSION

There are at least two important reasons for considering related FT's :

1) While $C(w)$ working on a word $x \in F(w)$ outputs position of beginning of the first occurrence of x in w, $C(w^R)$ working on x^R outputs position of the end of the last occurrence of x in w. So the related transducers can be used to examine multiple occurancies of a given factor x in word w. More precisely, the string $z \in F(x)$ has multiple occurrence in x iff

$$pos(z,x) + pos(z^R,x^R) < |x| - |z| \quad .$$

2) If we are interested only in finding some occurrence of x in w (not necessarily first), $C(w^R)$ working on x^R will bring the same profit as $C(w)$ working on x. Moreover, $C(w^R)$ can be sometimes much smaller than $C(w)$ as pointed out in Section 3.

ACKNOWLEDGEMENT

Many thanks are addressed to Professore Maxime Crochemore, University of Paris-Nord, for his valuable comments on the ideas presented above.

SOUHRN

PŘÍBUZNÉ TRANSDUCERY

PETR LISOŇĚK

V návaznosti na definici (faktorového) transduceru C(w) slova w jakožto jistého Mealyho stroje operujícího nad všemi podslovy slova w, podanou v [1], je v našem článku uvedena myšlenka studovat společně "příbuzné" transducery C(w) a $C(w^R)$ pro pevné slovo w. Jsou uvedeny dvě aplikace tohoto přístupu - snížení počtu stavů transduceru a jednoduchý algoritmus nalezení opakovaných výskytů daného podslova ve slově w.

РЕЗЮМЕ

РОДСТВЕННЫЕ ТРАНСДУЦЕРЫ

П. ЛИСОНЕК

Приведенная статья относится к понятию факторного трансдуцера $C(w)$ из статьи /1/. В нашей статье предложена идея изучать вместе "родственные" трансдуцеры $C(w)$ и $C(w^R)$ из-за /по меньшей мере/ двух аппликаций: уменьшение трансдуцеров и искание повторяющихся появлений некоторого фактора в слове w.

REFERENCES

[1] C r o ch e m o r e, M.: Transducers and repetitions. Theoret. Comput.Sci. 45 (1986), 63-86.

[2] A h o, A.V. - C o r a s i c k, M.J.: Efficient string matching: An aid to bibliographic research, Comm.ACM 18 (1975), 333-340.

[3] M o r r i s, J.H. - P r a t t, V.R.: A linear pattern-matching algorithm, Tech.Rept.40, Comput.Center,Univ.of California, Berkeley, 1970.

[4] C r o ch e m o r e, M., personal communication. Winter 1989.

Author´s address:
RNDr.Petr Lisoněk
Katedra výpočetní techniky PřF UP
771 46 Olomouc
Czechoslovakia

Acta UPO, Fac.rer.nat. 97, Mathematica XXIX, 1990, 291 - 299.