Playing with patterns, searching for strings

Playing with patterns, searching for strings

Jaap van der Woude

87/13

june 87

COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science of Eindhoven University of Technology.
Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review.
Copies of these notes are available from the author or the editor.

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands

editor: F.A.J. van Neerven

Playing with patterns, searching for strings

Jaap van der Woude

june 87

§ 0

Pattern identification is a source of several instructive programming exercises. We shall present one such exercise that is especially interesting for problems dealing with periodicity. In particular it enables us to treat preprocessing and search in the Knuth-Morris-Pratt pattern search algorithm as a unity.

Some remarks on names and notations:

Let Σ be a fixed alphabet. A word over Σ, i.e. an element of the free monoid (Σ^*, Λ) generated by Σ, will be called a *string*. $\Sigma^* \setminus \{\Lambda\}$ will be denoted by Σ^+, the nonempty strings. (In the sequel, capitals refer to strings, lowercase letters to naturals (including 0) or functions, unless stated otherwise.)

Let X be a string. In order to facilitate references to the length $|X|$ of X and symbols occuring in X, we shall write $X\,(i : 0 \leq i < N)$. Then $|X| = N$ and $X(i)$ is the $i+1^{th}$ symbol of X.

If $n \leq N$ we shall write $X \downarrow n$ for $X\,(i : 0 \leq i < n)$, the prefix of X withe the length n.

A string X is called *periodic* if $X = P^m$ for some $P \in \Sigma^+$ and $m \geq 2$, where P^m is the concatenation of m copies of P. In that case, P as well as $|P|$ is called *a period* of X. *The period* of X is the smallest period (if any).

For strings X and Y

$X \leq Y$ denotes "X is a prefix of Y"
$X < Y$ means $X \leq Y \wedge X \neq Y$

So much for general remarks on names and notations.

First we shall state the basic problem (MPP) in its "historical" context, subsequently we shall give two applications

- Pattern-search (e.g. Knuth-Morris-Pratt)
- Periodicity-search (for all prefixes)

§ 1. The MPP problem

The problem we would like to consider evolved from the exercise below. We shall give some heuristics for this program evolution.

Exercise ("carrécheck")
A string X is a *carré* if $X = P^2$ for some string $P \in \Sigma^+$. Derive a program to find every prefix of X that is a carré.
The formal specification :

$|[\ N$: int ; $\{N \geq 1\}$
  $X\ (i : 0 \leq i < N)$: string ;
  $|[\ c\ (i : 1 \leq i \leq N)$: array of bool ;
    CARRÉCHECK
    $\{(\mathbf{A}\ i : 1 \leq i \leq N : c(i) \equiv \mathrm{car}(X \downarrow i))\}$
  $]|$
$]|$

where

(0) $\qquad \mathrm{car}(X) \equiv (\mathbf{E}\ P : P \in \Sigma^+ : X = PP)$.

A standard feature in programming methodology is: weaken in the postcondition by replacement of a constant by a variable.

If we consider the term in definition (0) : $X = PP$ we might think of P as being a constant. Symmetry tells us to replace P by a variable twice. So $X = PP$ might be "generalized" to $X = PE \wedge X = FP$ for variables E and F.
I.e. P is a pre- and postfix of X.
So for strings P and X we define

(1) $\qquad P\ \underline{\mathrm{pp}}\ X \equiv (\mathbf{E}\ E, F : E, F \in \Sigma^+ : X = PE \wedge x = FP)$

and note that $P\ \underline{\mathrm{pp}}\ X \wedge |X| = 2 * |P| \Rightarrow \mathrm{car}(X)$.

The lack of reflexivity of $\underline{\mathrm{pp}}$, created by the domain in (1), seems unnatural, but it results from the definition of periodicity, i.e. the domain in definition (0).
Moreover, incorporating reflexivity of $\underline{\mathrm{pp}}$ leads to additional non-triviality analysis (in $\underline{\mathrm{mpp}}$ below for example).

First we give a few properties of $\underline{\mathrm{pp}}$, the simple proofs are omitted. Let H, P and X be strings and $h, x \in \Sigma$, then

(2) $\qquad H\ \underline{\mathrm{pp}}\ P \wedge P\ \underline{\mathrm{pp}}\ X \Rightarrow H\ \underline{\mathrm{pp}}\ X$

(3) $\qquad H\ \underline{\mathrm{pp}}\ X \wedge P\ \underline{\mathrm{pp}}\ X \wedge |H| < |P| \Rightarrow H\ \underline{\mathrm{pp}}\ P$

(4) $\qquad Hh\ \underline{\mathrm{pp}}\ Xx \equiv Hh < Xx \wedge H\ \underline{\mathrm{pp}}\ X \wedge h = x$

Property 4 will be used for prefixes of a fixed string X:

(4') $\quad X \downarrow k+1 \ \underline{\text{pp}}\ X \downarrow n+1 \ \equiv \ X \downarrow k \ \underline{\text{pp}}\ X \downarrow n \wedge X(k) = X(n)$

(4'') $\quad \neg\, X \downarrow k+1 \ \underline{\text{pp}}\ X \downarrow n+1 \ \equiv \ \neg\, X \downarrow k \ \underline{\text{pp}}\ X \downarrow n \vee X(k) \neq X(n)$

By 2, the transitivity of pp, we feel invited to consider maximal pre- and postfixes. So define :

(5) $\quad P \ \underline{\text{mpp}}\ X \ \equiv \ P \ \underline{\text{pp}}\ X \wedge (\mathbf{A}\, H : H \ \underline{\text{pp}} X : H = P \vee H \ \underline{\text{pp}} P)$

Note that the following are equivalent

$\underline{a}$ $P \ \underline{\text{mpp}}\ X$
$\underline{b}$ $P \ \underline{\text{pp}}\ X \wedge (\mathbf{A}\, H : H \ \underline{\text{pp}} X : |H| \leq |P|)$
$\underline{c}$ $P \ \underline{\text{pp}}\ X \wedge (\mathbf{A}\, H : H \ \underline{\text{pp}} X : H \leq P)$

In section 6 we show that, given $P \ \underline{\text{mpp}}\ X$, we have

$$\text{car}(X) \ \equiv \ |X| \ \underline{\text{mod}}\ (2 * (|X| - |P|)) = 0$$

So indeed the generalization (weakening of carré) is fruitful. The carrécheck problem has evolved to the MPP *problem*:

Derive a program that calculates for every prefix of a given string its maximal pre- and postfix. The formal specification

```
|[N : int ; {N ≥ 1}
  X (i : 0≤ 1<N) : string ;
  |[ f (i : 1≤ i ≤ N) : array of [0..N) ;
     MPP
     {(A i : 1≤ i ≤N : X ↓f (i) mpp X ↓i)}
  ]|
]| .
```

§ 2. Solution to the MPP problem

Forced by the postcondition of MPP :

$R_0 \quad (\mathbf{A}\, i : 1 \le i \le N : X \downarrow f(i)\, \underline{\text{mpp}}\, X \downarrow i)$,

we choose the following invariants :

$P_0 \quad (\mathbf{A}\, i : 1 \le i \le n : X \downarrow f(i)\, \underline{\text{mpp}}\, X \downarrow i)$

$P_1 \quad 1 \le n \le N$.

Approximation 0 (for MPP)

$n := 1 \,;\, f:(1) = 0 \quad \{P_0 \wedge P_1\}$
; **do** $n \neq N \rightarrow S_0 \quad \{X \downarrow k \ \underline{\text{mpp}}\ X \downarrow n+1\}$
$\quad ;\, f : (n+1) = k \quad \{(P_0 \wedge P_1)^n_{n+1}\}$
$\quad ;\, n := n+1 \quad \{P_0 \wedge P_1\}$
od $\{P_0 \wedge P_1 \wedge n = N$, hence $R_0\}$

On cosmetical grounds, with 4' in mind, we consider a slightly different postcondition for S_0 :

$R_1 \quad X \downarrow k+1 \ \underline{\text{mpp}}\ X \downarrow n+1$

By definition of mpp (version 5*b*) and by 4' , R_1 equivales

$$X \downarrow k \ \underline{\text{pp}}\ X \downarrow n \ \wedge \ X(k) = X(n) \ \wedge \ (\mathbf{A}\, j : X \downarrow j \ \underline{\text{pp}}\ X \downarrow n+1 : j \le k+1)$$

This leads us to a repetition for S_0 with guard $X(k) \neq X(n)$ and invariants :

$Q_0 \quad (\mathbf{A}\, j : X \downarrow j \ \underline{\text{pp}}\ X \downarrow n+1 : j \le k+1)$

$Q_1 \quad X \downarrow k \ \underline{\text{pp}}\ X \downarrow n \ \wedge k \ge 0$

Approximation 1 (for S_0)

$k := f(n)$ $\{Q_0 \wedge Q_1$, see the note below$\}$
; **do** $X(k) \neq X(n) \wedge k \neq 0$
$\quad \rightarrow$ "decrease k under invariance of Q (and P)"
od $\{Q_0 \wedge Q_1 \wedge (X(k) = X(n) \vee k = 0)\}$

The second conjunct in the guard is forced upon us by the wish to decrease k, leaving $k \ge 0$ invariant.

note $\quad P_0$

$\Rightarrow$ {instantiation at n , def. mpp (5*b*)}
$\quad X \downarrow f(n)\ \underline{\text{pp}}\ X \downarrow n \wedge (\mathbf{A}\, j : X \downarrow j \ \underline{\text{pp}}\ X \downarrow n : j \le f(n))$
$\Rightarrow$ {by 4' : $X \downarrow j+1 \ \underline{\text{pp}}\ X \downarrow n+1 \ \Rightarrow X \downarrow j \ \underline{\text{pp}}\ X \downarrow n$}
$\quad X \downarrow f(n)\ \underline{\text{pp}}\ X \downarrow n \wedge (\mathbf{A}\, j : X \downarrow j+1 \ \underline{\text{pp}}\ X \downarrow n+1 : j \le f(n))$
$\Rightarrow$ {dummy change}
$\quad X \downarrow f(n)\ \underline{\text{pp}}\ X \downarrow n \wedge (\mathbf{A}\, j : X \downarrow j \ \underline{\text{pp}}\ X \downarrow n+1 : j \le f(n)+1)$
$=$ {def. Q_0, Q_1}
$\quad (Q_1 \wedge Q_0)^k_{f(n)}$

Under assumption of $Q_0 \wedge Q_1 \wedge P_0 \wedge P_1$ and the guard, we study reduction of k. First with respect to Q_0 :

Let $j > 0$ then

$$
\begin{array}{ll}
 & X\downarrow j \ \underline{\text{pp}}\ X\downarrow n+1 \\
\Rightarrow & \{Q_0 \text{ ; by 4" }, X(k) \neq X(n) \Rightarrow \neg\, X\downarrow k+1 \ \underline{\text{pp}}\ X\downarrow n+1\} \\
 & X\downarrow j \ \underline{\text{pp}}\ X\downarrow n+1 \wedge j \leq k+1 \wedge j \neq k+1 \\
\Rightarrow & \{j \geq 0, 4'\} \\
 & X\downarrow j-1 \ \underline{\text{pp}}\ X\downarrow n \wedge j-1 < k \\
\Rightarrow & \{Q_1, 3\} \\
 & X\downarrow j-1 \ \underline{\text{pp}}\ X\downarrow k \\
\Rightarrow & \{P_0, 0 \neq k < n \text{ , def. } \underline{\text{mpp}}\ 5a\} \\
 & j-1 \leq f(k)
\end{array}
$$

For $j = 0$, certainly we have $j \leq f(k)+1$. Hence $Q_0{}^{k}_{f(k)}$ holds.

With respect to Q_1 :

$$
\begin{array}{ll}
 & P_0 \wedge Q_1 \\
\Rightarrow & \{k \neq 0 : P_0 \text{ instantiated at } k\} \\
 & X\downarrow f(k) \ \underline{\text{pp}}\ X\downarrow k \wedge X\downarrow k \ \underline{\text{pp}}\ X\downarrow n \\
\Rightarrow & \{2\} \\
 & X\downarrow f(k) \ \underline{\text{pp}}\ X\downarrow n \\
= & \{\text{def. } Q_1, f(k) \geq 0\} \\
 & Q_1{}^{k}_{f(k)}
\end{array}
$$

This shows that "decrease k under invariance ..." is established by $k := f(k)$. (Certainly $P_0 \wedge P_1$ is not affected.)

Because of the conjunct $k \neq 0$ in the guard, neither R_1 not the original postcondition of S_0 are met, but a mixture is :

for $\quad Q_0 \wedge Q_1 \wedge X(k) = X(n) \Rightarrow R_1$

and $\quad Q_0 \wedge Q_1 \wedge X(k) \neq X(n) \wedge k = 0$

$$
\begin{array}{ll}
\Rightarrow & \{4"\} \\
 & Q_0 \wedge Q_1 \wedge \neg\, X\downarrow k+1 \ \underline{\text{pp}}\ X\downarrow n+1 \wedge k = 0 \\
\Rightarrow & \{X\downarrow 0 \ \underline{\text{pp}}\ X\downarrow n+1, \text{ def. } \underline{\text{mpp}}\} \\
 & X\downarrow k \ \underline{\text{mpp}}\ X\downarrow n+1 \wedge k = 0
\end{array}
$$

This proves the following solution for MPP :

$n := 1$; $f:(1) = 0$ $\{P_0 \wedge P_1\}$
; **do** $n \neq N$
→ |[k : int ;
$k := f(n)$ $\{Q_0 \wedge Q_1\}$
; **do** $X(k) \neq X(n) \wedge k \neq 0$
→ $k := f(k)$
od $\{Q_0 \wedge Q_1 \wedge (X(k) = X(n) \vee k = 0)\}$
; **if** $X(k) = X(n) \rightarrow$ $\{X \downarrow k+1$ mpp $X \downarrow n+1\}$ $k := k+1$
[] $X(k) \neq X(n) \rightarrow$ $\{X \downarrow k+1$ mpp $X \downarrow n+1\}$ skip
fi $\{X \downarrow k$ mpp $X \downarrow n+1\}$
; $f : (n+1) = k$ $\{(P_0 \wedge P_1)^n_{n+1}\}$
]|
; $n := n+1$
od $\{P_0 \wedge P_1 \wedge n = N$, hence $R_0\}$

For the complexity of the algorithm, consider k to exist outside the innerblock $(P_2: k = f(n))$. A variant function that shows linearity is $2N - 2n + k$.

§ 3. Pattern search

Let $P \in \Sigma^+$ be fixed, the pattern. Suppose we are interested in (all) occurrences of P in a string Z.

Put $Y = PZ$, then we are searching for numbers n, such that

$$n \geq 2 * |P| \wedge P \ \underline{\text{pp}}\ Y \downarrow n$$

In this setting the pattern might be represented by its length only.

Let X be a string, p a number $1 \leq p < |X|$.
As $X \downarrow p$ is a postfix of $X \downarrow n$ iff $p = n \vee X \downarrow p \ \underline{\text{pp}}\ X \downarrow n$, we define (the occurrence of the pattern as postfix of $X \downarrow n$):

(6) $$\text{O(n)} \equiv p = n \ \vee \ X \downarrow p \ \underline{\text{pp}}\ X \downarrow n$$

Let f be as in the MPP problem. It seems reasonable to hope for suitable O-information in the mpp-knowledge recorded in f. Indeed, for strings H, P and X we have

(7) $$P \ \underline{\text{mpp}}\ X \ \Rightarrow (H \ \underline{\text{pp}} X \ \equiv H = P \vee H \ \underline{\text{pp}} P)\ .$$

Property 7, which is closely linked to 3, follows easily from 5*a*, 2.
It relates O(n) to $f(n)$ as follows :

O(n)
= {6}
$p = n \vee X \downarrow p \ \underline{\text{pp}}\ X \downarrow n$
= {$X \downarrow f(n) \ \underline{\text{mpp}}\ X \downarrow n$, 7 with $H, P, X := X \downarrow p, X \downarrow f(n), X \downarrow n$}
$p = n \vee X \downarrow p = X \downarrow f(n) \vee X \downarrow p \ \underline{\text{pp}}\ X \downarrow f(n)$
= {$X \downarrow p = X \downarrow f(n) \ \equiv \ p = f(n)$; 6}
$p = n \vee \text{O}(f(n))$

As $f(n) < n$ (nonreflexivity of pp), O(n) depends only on $f(n)$ and O(i : $1 \leq i < n$). This settles pattern search as a simple extension of the MPP problem, by adding invariant

P_3 $\quad (\mathbf{A}\ i : 1 \leq i \leq n : o(n) \equiv \text{O(n)})$

and initialization $o : (1) = (p = 1)$ $\{P_3\}$
extra statement ; $o : (n+1) = (p = n+1) \vee o(f(n+1))$ $\{P_{3n+1}^n\}$
immediately following the innerblock.

§ 4. Knuth - Morris - Pratt

The pattern search presented in the previous section has a serious drawback : storage linear in the length of the given string (concatenated with the pattern). Indeed, the algorithm needs $f(i : 1 \leq 1 \leq n)$ to calculate $f(n+1)$ and O(i : $1 \leq i \leq n$) and $f(n+1)$ to calculate O(n+1).
As, for fixed p, we are interested in n such that $X \downarrow p$ $\underline{pp}$ $X \downarrow n$ (instead of $\underline{mpp}$!) the information recorded in f exceeds our needs : we might do with pre- and postfixes with lengths at most p. So we define : (P and p are not related !)

(8) $$P \ \underline{\pi\pi} \ X \ \equiv \ P \ \underline{pp} \ X \wedge \mid P \mid < p$$

(9) $$P \ \underline{\mu\pi\pi} \ \equiv \ P \ \underline{\pi\pi} \ X \wedge (\mathbf{A}\, H : H \ \underline{\pi\pi} X : H = P \vee H \ \underline{\pi\pi} P)$$

The reader is urged to convince himself of the truth of the $\underline{\pi\pi}$-versions of $2, 3, 5^{a,b,c}$ (i.e. only ($\underline{m}$)$\underline{pp}$ replaced by ((μ) $\underline{\pi\pi}$). Property 4, however, has a slightly different $\underline{\pi\pi}$-version.
We shall only provide the $\underline{\pi\pi}$-version of 4' :

(10') $$X \downarrow k \ \underline{\pi\pi} \ X \downarrow n \wedge X(k) = X(n)$$

$$\equiv (X \downarrow k+1 \ \underline{\pi\pi} \ X \downarrow n+1 \wedge k < p-1) \vee (X \downarrow k+1 \ \underline{pp} \ X \downarrow n+1 \wedge k = p-1)$$

The $\mu\pi\pi$ problem is given by the postcondition

ρ_0 $\quad (\mathbf{A}\, i : 1 \leq i \leq N : X \downarrow \phi(i) \ \underline{\mu\pi\pi} \ X \downarrow i)$

For the solution of the $\mu\pi\pi$ problem we define invariants, (the obvious adaptations of the invariants for *MPP*)

π_0 $\quad (\mathbf{A}\, i : 1 \leq i \leq n : X \downarrow \phi(i) \ \underline{\mu\pi\pi} \ X \downarrow i)$

π_1 $\quad 1 \leq n \leq N$

ψ_0 $\quad (\mathbf{A}\, j : X \downarrow j \ \underline{\pi\pi} X \downarrow n+1 : j \leq k+1)$

ψ_1 $\quad X \downarrow k \ \underline{\pi\pi} \ X \downarrow n \wedge k \geq 0$

Except from the obvious adaptations, $\mu\pi\pi$ differs from MPP only in the case analysis in the innerblock : the drawback of 10' :

Certainly,

$$\psi_0 \wedge \psi_1 \wedge X(k) = X(n) \wedge k < p-1$$
$$\Rightarrow \{10', \text{def. } \underline{\mu\pi\pi}\}$$
$$X \downarrow k+1 \ \underline{\mu\pi\pi} \ X \downarrow n+1$$

but
$$\psi_0 \wedge \psi_1 \wedge X(k) = X(n) \wedge k = p-1$$
$$\Rightarrow \{10', k+1 = p\}$$
$$X \downarrow p \ \underline{pp} \ X \downarrow n+1$$
$$\Rightarrow \{\pi_0, \text{instantation at } p \ ; 2\}$$
$$X \downarrow \phi(p) \ \underline{\mu\pi\pi} \ X \downarrow n+1$$

This shows that the alternative statement following the inner repetition should be changed (for the $\mu\pi\pi$ problem) to

$$
\begin{array}{l}
\{\psi_0 \wedge \psi_1 \wedge (X(k)=X(n) \vee k=0)\} \\
\;; \textbf{if } X(k)=X(n) \wedge k<p-1 \rightarrow \{X\downarrow k+1 \;\underline{\mu\pi\pi}\; X\downarrow n+1\}\; k:=k+1 \\
\;\;[]\; X(k)=X(n) \wedge k=p-1 \rightarrow \{X\downarrow \phi(p) \;\underline{\mu\pi\pi}\; X\downarrow n+1\}\; k:=\phi(p) \\
\;\;[]\; X(k)\neq X(n) \qquad\qquad \rightarrow \{k=0 \wedge X\downarrow k \;\underline{\mu\pi\pi}\; X\downarrow n+1\} \text{ skip} \\
\textbf{fi} \\
\{X\downarrow k \;\underline{\mu\pi\pi}\; X\downarrow n+1\}
\end{array}
$$

This and the change from f to ϕ make the solution of MPP to a solution of $\mu\pi\pi$. The code will reappear in the Knuth-Morris-Pratt pattern search algorithm, so we shall leave it with this. Note that for calculation of $\phi(n+1)$ only $\phi(n)$ and $\phi(i : 1\leq i\leq p)$ are needed : $\phi(n)$ for initializing k, $\phi(i : 1\leq i\leq p)$ in the inner repetition. I.e. $\mu\pi\pi$ needs storage proportional to the "pattern length".

Similar to § 3. we now transform $\mu\pi\pi$ to a pattern search algorithm : Knuth-Morris-Pratt.
With respect to occurrence of $X\downarrow p$ as postfix of $X\downarrow n+1$, note that

$$
\begin{array}{ll}
 & O(n+1) \\
= & \{6\} \\
 & \quad p=n+1 \vee X\downarrow p \;\underline{pp}\; X\downarrow n+1 \\
= & \{4'; \text{def. } \underline{\pi\pi}\} \\
 & \quad p=n+1 \vee (X\downarrow p-1 \;\underline{\pi\pi}\; X\downarrow n \wedge X(p-1)=X(n)) \\
= & \{\text{def. of } \underline{\pi\pi} : \psi_{0p-1}^{k} \equiv \textbf{true} ; \text{def. } \psi\} \\
 & \quad p=n+1 \vee (\psi_0 \wedge \psi_1 \wedge X(k)=X(n))_{p-1}^{k}
\end{array}
$$

So calculation of $O(n+1)$ depends only on the postcondition of the inner repetition and occurrence is to be signalled in the second alternative (the occurrence at $n+1=p$ is not relevant of course).

We are now ready for the Knuth-Morris-Pratt algorithm.
As only $\phi(i : 1\leq i\leq p)$ and $\phi(n)$ are needed to calculate $\phi(n+1)$, we have to distinguish between

preprocessing – "filling ϕ"

search – "signalling occurrences".

This seperation of the two parts is inevitable, but an earlier separation is unnecessary, unelegant and confusing. In order to account for the reduced domain of ϕ we modify π_0 to π_0^1, to "buffer" $\phi(n)$ we add π_2

π_0^1 $(A\, i : 1\leq i\leq p \;\underline{\min} n : X\downarrow\phi(i) \;\underline{\mu\pi\pi}\; X\downarrow i)$

π_2 $X\downarrow k \;\underline{\mu\pi\pi}\; X\downarrow n$

and we take k outside the inner repetition.

The Knuth-Morris-Pratt pattern search algorithm we derived :

$|[\ k$: int ;
$\phi(i : 1 \le i \le p)$: array of $[0..p-1)$;
$n, k := 1, 0 ; \phi :(1) = 0\ \{\pi_0^1 \wedge \pi_1 \wedge \pi_2\}$
; **do** $n \neq N$
$\rightarrow\ \{\pi_0^1 \wedge \pi_1 \wedge \pi_2 \wedge n \neq N$, so $\psi_0 \wedge \psi_1\}$
do $X(k) \neq X(n) \wedge k \neq 0$
$\rightarrow k := \phi(k)$
od $\{\psi_0 \wedge \psi_1 \wedge (X(k) = X(n) \vee k = 0)\}$
; **if** $X(k) = X(n) \wedge k < p-1 \rightarrow k := k+1$
$[]\ X(k) = X(n) \wedge k = p-1 \rightarrow k := \phi(p)$; "*MATCH*"
$[]\ X(k) \neq X(n) \rightarrow$ skip
fi $\{X \downarrow k\ \underline{\mu\pi\pi}\ X \downarrow n+1$, so $\pi_{2n+1}^n\}$
; **if** $n < p \rightarrow \phi : (n+1) = k$
$[]\ n \geq p \rightarrow$ skip
fi $\{(\pi_0^1 \wedge \pi_1 \wedge \pi_2)_{n+1}^n\}$
; $n := n+1$
od $\{\pi_0^1 \wedge \pi_1 \wedge \pi_2 \wedge n = N$, so $\rho_0\}$
$]|$

The interested reader might want to try a direct approach via $\mu\pi\pi$.

§ 5. Further remarks on pattern search

The second alternative statement in the algorithm above, distinguishing preprocessing and search, may also lead to a code with two (sequential) repetitions, one for each alternative. We chose for the form above to stress the uniformity: the difference between the parts is solely based upon coding, the genesis doesn't differentiate!

Several people noticed the strong resemblance of those parts, but in the literature we searched in vain for a presentation or derivation (at all) of the algorithm that did justice to that resemblance. ([C 85] and [W 86] deserve some credit).

[Note that even in 1983 the preprocessing was said to be "complicated and difficult to understand" ([S 83] p. 242). As the two parts are almost identical such a statement is puzzling. Has it anything to do with the widespread chaotic algorithm presentation? (e.g. [KMP 77], [BM 77])].

In our opinion, exploitation of pre- and postfixes simplified the "derivation" of the algorithm such that it becomes within reach of every freshmen course.

We conclude the discussion of pattern search with a remark on the Boyer-Moore fast pattern search ([BM 77]). As this algorithm is slightly beyond the scope of this paper, we shall only hint at its relation with the MPP problem.

Consider $X \in \Sigma^*$ and pattern $X\downarrow p$. In the Knuth-Morris-Pratt pattern search we decided to build up pre- and postfixes bit by bit, but we could have been greedier :
To that end consider the (linear-search-like) invariant

PBM $\quad (\mathbf{A}\ i : X\downarrow p \ \underline{\text{pp}}\ X\downarrow i : i > n)$

As $X\downarrow p \ \underline{\text{pp}}\ X\downarrow n+1 \ \Rightarrow \{4'\}\ X(p-1) = X(n)$ we <u>first</u> check $X(n)$ as a candidate for the end of a pattern occurrence.
Let $s = (\mathbf{MAX}\ j : 0 \le j \le p \wedge X(n) = X(j) : j)\ \underline{\text{max}}\ -1$
Then PBM^{n}_{n+p-s} holds.
In other words : the first candidate m to satisfy $X\downarrow p \ \underline{\text{pp}}\ X\downarrow m$ is $m = n + p - s$.
If $s < p - 1$ we can "leap further", if $s = p - 1$ we check $X(n-1)$, etcetera.
This requires knowledge of the occurrences of values and periodicities in the pattern.
The reader is challenged to give a "derivation" of the Boyer-Moore fast pattern search based on this early deviation of the MPP problem.

§ 6. Periodicity search

In section 1 we "generalized" the carrécheck exercise to the MPP problem, and we promised to show that a solution for carrécheck is found as soon as MPP is solved.
We shall keep our promise in the following way :

- we give a variant of carrécheck
- we proclaim an enrichment of MPP that solves the (variant) exercise
- we perform some string-mathematics to prove that the exercise is solved by that enrichment of MPP.

For fixed $m \geq 2$ consider the following postcondition

R $\quad (\mathbf{A}\, i : 1 \leq i \leq N : c(i) \equiv (\mathbf{E}\, P : P \in \Sigma^+ : X \downarrow i = P^m)$

$\wedge\ (\mathbf{A}\, i : 1 \leq i \leq N : \mathrm{per}(i) = i \ \underline{\min}\ (\mathbf{MIN} p : p \text{ period of } X \downarrow i : p))$

In case $m = 2$, the first conjunct of R is just the postcondition of carrécheck.
The second conjuct of R means :
$\mathrm{per}(i)$ is the period of $X \downarrow i$ if $X \downarrow i$ is periodic, otherwise $\mathrm{per}(i) = i$.
Obviously we should extent invariant P for the MPP problem with a conjunct P_4 to get an invariant for the new problem.

$P_4 \quad R_n^N$

Initialization of P_4 : ; $c : (1) =$ **false** ; per: $(1) = 1$.
The outer repetition should contain an establishment of P_{4n+1}^n .
So, following "$f : (n+1) = k$ $\{P_{0n+1}^n\}$" and before "$n := n+1$", we proclaim the statement list :

; $c : (n+1) = (n+1)$ mod $(m * (n+1-f(n+1))) = 0$
; **if** $(n+1)$ mod $(n+1-f(n+1)) = 0 \rightarrow$ per: $(n+1) = n+1-f(n+1)$
[] $(n+1)$ mod $(n+1-f(n+1)) \neq 0 \rightarrow$ per: $(n+1) = n+1$
fi $\{P_{4n+1}^n$, see corollary 5 to follow $\}$.

Indeed a minor adaptation, but it takes a proof!

The string-mathematics to follow is quite elementary, and has nothing to do with programming and - methodology. So we adopt a more conventional mathematical style, but (for the convenience of non-mathematicians) we still take small steps in the proofs.
The basic idea is to squeeze periodicity information out of pp or mpp knowledge.

Let D , $Y \in \Sigma^+$ with D pp Y. Then there are E , $F \in \Sigma^+$ such that $Y = DE \wedge Y = FD$.
Lemmata 1 and 2 tell us about (almost) periodicity of Y. (they are well-known, e.g. see [L 79] Ch. 11.5). Much more can be said if D mpp Y, some of which is done in 3,4,5.

Lemma 1 Let $D \in \Sigma^*$ and $E, F \in \Sigma^+$ such that $DE = FD$.
Then there are $L \in \Sigma^*$, $K \in \Sigma^+$ and $n \geq 0$ with

0 $D = F^n L$ and $L < F$ (hence $DE = FD = F^{n+1} L$)

1 $E = KL$ and $F = LK$.

Proof Certainly, there are $L \in \Sigma^*$ and $n \geq 0$ such that $D = F^n L$ and $|L| < |F|$.
Then $F^n LE = DE = FD = F^{n+1} L$, so $LE = FL$.
As $|L| < |F| = |E|$, there are $K, K^1 \in \Sigma^+$ with $LK = F$ and $E = K^1 L$.
Hence, $L(K^1 L) = LE = FL = (LK)L$ and it follows that $K^1 = K$. □

Lemma 2 Let $D, F \in \Sigma^*$ with $DF = FD$. Then there is a $P \in \Sigma^*$ such that $D, F \in \{P^m \mid m \geq 0\}$. If $D, F \in \Sigma^+$ it follows that DF is periodic with period at most $gcd(|D|, |F|)$.

Proof By induction to the length of DF.
If $D = \Lambda$ or $F = \Lambda$ the existence of P is obvious.
Let $D, F \in \Sigma^+$. By 1, there are K, L, n such that $D = F^n L$, $F = KL$ and $F = LK$.
Hence $KL = LK$.
As $D \neq \Lambda$, $|KL| = |F| < |DF|$, so by induction there is a $P \in \Sigma^*$ (as $K \in \Sigma^+$ even $P \in \Sigma^+$) such that $K, L \in \{P^m \mid m \geq 0\}$.
Consequently, $D, F \in \{P^m \mid m \geq 0\}$ which proves the first part.
If $D, F \in \Sigma^+$ then $DF \in \{P^{m+2} \mid m \geq 0\}$ while $P \in \Sigma^+$.
Note that $|P|$ divides $|D|$ and $|F|$. □

Lemma 3 Let $Y = FD$ and D mpp Y. Then F is not periodic.

Proof By definition of (m)pp, $F \neq \Lambda$. So, by 1, there are L, n with $D = F^n L$ and $L < F$.
Suppose F is periodic, say $F = Q^m$ for some $Q \in \Sigma^+, m \geq 2$.
Then $QF = FQ$, hence $QL < QF = FQ$. As also $L < F < FQ$, it follows that FQ has both L and QL as prefix. Since $|L| < |QL|$ we have $L < QL$ and, equivalently, $Q^m Q^{mn-1} L < Q^m Q^{mn} L$.
As $Y = FD = Q^m Q^{mn} L$, it follows that $Q^m Q^{mn-1} L$ pp Y. However, since $m \geq 2$, $|D| = |Q^{mn} L| < |Q^m Q^{mn-1} L|$ which contradicts D mpp Y. This falsifies periodicity of F. □

Lemma 4 Let $Y = FD$, D mpp Y . Let P pp Y and $|P| \geq |F|$, then there is a $k \geq 0$ such that $D = F^k P$. (I.e. all pre-postfixes of Y with lenght $\geq |F|$ are known).

Proof As D mpp Y , $F \neq \Lambda$, so there are $n \geq 0$ and $L < F$ with $D = F^n L$.
Because $|L| < |F| \leq |P|$ and P is a postfix of D , L is a postfix of P. Hence there are a $k : 0 \leq k \leq n$ and $H < F$ with $D = F^k HP$. Let $HG = F$ then $P = GF^{n-k-1} L$.
As $|H| + |G| = |F| \leq |P| = |G| + |F^{n-k-1} L|$, $|H| \leq |F^{n-k-1} L|$.
Since $H < F^{n-k}$ and $F^{n-k-1} L < F^{n-k}$ it follows that $H \leq F^{n-k-1} L$, so $GH \leq GF^{n-k-1} L = P$.
On the other hand $HG = F \leq P$, so $GH = HG$.
As $H < F$, and, by 3, F is not periodic it follows from 2 that $H = \Lambda$, which shows $D = F^k P$.
□

Corollary 5 Let D mpp Y , say $Y = FD$. Let $m \geq 2$, then

$$(\mathbf{E}\, C : : Y = C^m) \text{ iff } |Y| \bmod (m * |F|) = 0 .$$

In particular, Y is a carré iff $|Y| \bmod 2|F| = 0$, and Y is periodic iff $D \neq \Lambda$ and $|Y| \bmod |F| = 0$.

Proof By 1, $Y = F^{n+1} L$ and $L < F$, so the if-part is obvious.
Let $Y = C^m$; note that since $m \geq 2$, C^{m-1} pp Y . As $|D| \geq |C^{m-1}|$, $|F| \leq |C| \leq |C^{m-1}|$, so by 4, $D = F^k C^{m-1}$.
Hence $F^{k+1} C^{m-1} = Y = CC^{m-1}$ and $C = F^{k+1}$, so $Y = F^{m*(k+1)}$.
The remark on Y being a carré is an instantation for $m = 2$.
Finally, as $Y \neq F$, $|Y| \bmod |F| = 0 \equiv (\mathbf{E}\, m : m \geq 2 : |Y| \bmod (m * |F|) = 0)$. □

Note that if Y is periodic, $|Y| - |D|$ is the period.

Acknowledgements

The carré problem and related other problems were "en vogue" in the environment of the Eindhoven University of Technology. So, inevitably, I was contaminated too.

I would like to thank all colleagues that contributed to the genesis of this paper by comments and interest.

References

[BM 77] Boyer, R.S. and Moore J.S.,
A fast string searching algorithm,
CACM 20 , 762 - 772 (1977).

[C 85] Chengdian, C.,
A derivation of the Knuth-Morris-Pratt pattern matching program,
EUT-report 85-Wsk-02,
Eindhoven University of Technology (1985).

[KMP 77] Knuth, D.E., Morris, J.H. and Pratt, V.R.,
Fast pattern matching in strings,
SIAM J. Comput. 6 , 323 - 350 (1977).

[L 79] Lallement, G.,
Semigroups and combinatorial applications,
Wiley-interscience, (1979).

[S 83] Sedgewick, R.,
Algorithms,
Addison - Wesley, (1983).

[W 86] Wiltink, G.,
Knuth-Morris-Pratt, private communication,
(1986).

In this series appeared :

No.	Author(s)	Title
85/01	R.H. Mak	The formal specification and derivation of CMOS-circuits
85/02	W.M.C.J. van Overveld	On arithmetic operations with M-out-of-N-codes
85/03	W.J.M. Lemmens	Use of a computer for evaluation of flow films
85/04	T. Verhoeff H.M.J.L. Schols	Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate
86/01	R. Koymans	Specifying message passing and real-time systems
86/02	G.A. Bussing K.M. van Hee M. Voorhoeve	ELISA, A language for formal specifications of information systems
86/03	Rob Hoogerwoord	Some reflections on the implementation of trace structures
86/04	G.J. Houben J. Paredaens K.M. van Hee	The partition of an information system in several parallel systems
86/05	Jan L.G. Dietz Kees M. van Hee	A framework for the conceptual modeling of discrete dynamic systems
86/06	Tom Verhoeff	Nondeterminism and divergence created by concealment in CSP
86/07	R. Gerth L. Shira	On proving communication closedness of distributed layers

86/08	R. Koymans R.K. Shyamasundar W.P. de Roever R. Gerth S. Arum Kumar	Compositional semantics for real-time distributed computing (Inf. & Control 1987)
86/09	C. Huizing R. Gerth W.P. de Roever	Full abstraction of a real-time denotational semantics for an OCCAM-like language
86/10	J. Hooman	A compositional proof theory for real-time distributed message passing
86/11	W.P. de Roever	Questions to Robin Milner - A responders commentary (IFIP86)
86/12	A. Boucher R. Gerth	A timed failures model for extended communicating processes
86/13	R. Gerth W.P. de Roever	Proving monitors revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)
86/14	R. Koymans	Specifying passing systems requires extending temporal logic
87/01	R. Gerth	On the existence of a sound and complete axiomatizations of the monitor concept
87/02	Simon J. Klaver Chris F.M. Verberne	Federatieve Databases
87/03	G.J. Houben J. Paredaens	A formal approach to distributed information systems
87/04	T. Verhoeff	Delay-insensitive codes - An overview
87/05	R. Kuiper	Enforcing non-determinism via linear time temporal logic specification

87/06	R. Koymans	Temporele logica specificatie van message passing en real-time systemen (in Dutch)
87/07	R. Koymans	Specifying message passing and real-time systems with real-time temporal logic
87/08	H.M.J.L. Schols	The maximum number of states after projection
87/09	J. Kalisvaart L.R.A. Kessener W.J.M. Lemmens M.L.P van Lierop F.J. Peters H.M.M. van de Wetering	Language extensions to study structures for raster graphics
87/10	T. Verhoeff	Three families of maximally nondeterministic automata
87/11	P. Lemmens	Eldorado ins and outs. Specifications of a data base management toolkit according to the functional model
87/12	K.M. van Hee A. Lapinski	OR and AI approaches to decision support systems
87/13	J. van der Woude	Playing with patterns, searching for strings