Squaring a cyclic permutation in situ

In this note we describe the development of an in-situ algorithm for squaring what in traditional mathematics is called a cyclic permutation, but what we call a ring. The development is guided by a calculus of rings, which has been designed precisely for the treatment of this and similar algorithms.

Notation and nomenclature

We consider a finite non-empty domain. Elements of the domain are indicated by small letters, sequences of elements by capitals. Concatenation is denoted by juxtaposition and empty is the empty sequence.

For this domain we introduce the notions ring and mingling of rings :

- for each non-empty sequence X of distinct elements, [X] (pronounced "ring X") is a ring ; the elements of [X] are the elements of X
- each ring is a mingling of rings ; for minglings h and k of rings, h m k is a mingling of rings (m, the mingling operator, is pronounced "mingled").

Specification of the algorithm

For a ring h we want to construct a program square satisfying functional specification

(0) $\{h = [U]\}$ square $\{h = [U] \underline{m} [U]\}$.

If minglings of rings and the mingling operator were primitives of our program notation, the statement $h := h \underline{m} h$ would do. They aren't, however, so before embarking on the development of an algorithm we first give some properties of minglings of rings.

Ring calculus

For rings and their minglings we postulate a number of rules.

R0: $[X Y] = [Y X]$ for disjoint (sequences) X and Y, the rule of rotation.

R1: $\underline{m}$ is associative.

R2: $\underline{m}$ is symmetric for disjoint rings, i.e. for rings without common element.

R3: $[X p] \underline{m} [p Y] = [X p Y]$ for disjoint X, p, and Y, the one-pivot rule; p is the pivot.

R4: $[p q] \underline{m} [p q] = [p] \underline{m} [q]$.

These rules together with the rules R5 through R8, derived below, are all we need from ring calculus.

R5: $[X p] \underline{m} [p] = [X p]$,

following from R3 with $Y :=$ empty.

R6: $[X p Y q] \underline{m} [p q] = [X p] \underline{m} [Y q]$,

following from

$[X_p Y_q] \underline{m} [p\,q]$

$=$ {R3 on the first term, pivot p; R1, associativity of $\underline{m}$, not to be mentioned any more in the sequel}

$[X_p] \underline{m} [p\,Y\,q] \underline{m} [p\,q]$

$=$ {R0 on the last two terms}

$[X_p] \underline{m} [Y\,q\,p] \underline{m} [q\,p]$

$=$ {R3 on the middle term, pivot q}

$[X_p] \underline{m} [Y\,q] \underline{m} [q\,p] \underline{m} [q\,p]$

$=$ {R4 on the last two terms}

$[X_p] \underline{m} [Y\,q] \underline{m} [q] \underline{m} [p]$

$=$ {R2}

$[X_p] \underline{m} [p] \underline{m} [Y\,q] \underline{m} [q]$

$=$ {R5 on the first two terms and on the last two terms}

$[X_p] \underline{m} [Y\,q]$.

R7: $[X_p] \underline{m} [Y\,q] \underline{m} [p\,q] = [X_p\,Y\,q]$,

following from

$[X_p] \underline{m} [Y\,q] \underline{m} [p\,q]$

$=$ {R6 on the first two terms}

$[X_p\,Y\,q] \underline{m} [p\,q] \underline{m} [p\,q]$

$=$ {R4 on the last two terms}

$[X_p\,Y\,q] \underline{m} [p] \underline{m} [q]$

$=$ {R5, twice}

$[X_p\,Y\,q]$

R8: $[X_p\,Y\,q\,Z\,r] \underline{m} [p\,q\,r] = [Z\,r\,Y\,q\,X\,p]$,

following from

$[X_p\,Y\,q\,Z\,r] \underline{m} [p\,q\,r]$

$=$ {R0 on the first term; R3 on the last term, pivot q}

$$
\begin{array}{ll}
 & [Z_r X_p Y_q] \ \underline{m} \ [p q] \ \underline{m} \ [q r] \\
= & \quad \{R6 \text{ on the first two terms}\} \\
 & [Z_r X_p] \ \underline{m} \ [Y q] \ \underline{m} \ [q r] \\
= & \quad \{R0 \text{ on the two outer terms}\} \\
 & [X_p Z_r] \ \underline{m} \ [Y q] \ \underline{m} \ [r q] \\
= & \quad \{R7 \text{ with } X.p := X_p Z_. r\} \\
 & [X_p Z_r Y_q] \\
= & \quad \{R0\} \\
 & [Z_r Y_q X_p] \ .
\end{array}
$$

Representation

Minglings of rings not being available primitives of our program notation, we choose a mingling h of disjoint rings and an array H to be coupled by the representational convention

(1) for each element r and for all sequences X and Y such that $[X r Y]$ is a ring of h,
$H.r$ = the first element of sequence YXr.

(In other words, for each element r occurring in a (i.e. precisely one) ring of h, $H.r$ is r's "right successor" in that ring.)

Note that in the above convention the disjointness of the rings in h is essential. Rules R5 through R8 from ring calculus tell us that minglings with small rings preserve disjointness, which enables us to express these transformations in terms of H. For the transformation

(2) $\{h = [X_p]\}\ h := h \ \underline{m} \ [p] \ \{h = [X_p] \ .$ from R5$\}$,

apparently a skip suffices.

For the transformation

(3) $\{ h = [Xp\ Yq] \}$
$h := h\ \underline{m}\ [p\ q]$
$\{ h = [Xp]\ \underline{m}\ [Yq]$, from R6$\}$

we observe that its precondition implies (see (1)))

H.p = the first element of sequence Yq
H.q = the first element of sequence Xp ,

and that its postcondition requires that

H.p = the first element of sequence Xp
H.q = the first element of sequence Yq ,

while for the elements of X and Y , H does not change. Hence, the transformation is accomplished by H: swap.(p, q) .
For the transformation

$\{ h = [Xp]\ \underline{m}\ [Yq] \}$
$h := h\ \underline{m}\ [p\ q]$
$\{ h = [Xp\ Yq]$, from R7$\}$

we observe that it is the inverse of (3), and -hence- that it can be accomplished by the inverse of H: swap. (p, q) , i.e. by H: swap.(p, q) .

As a <u>corollary</u> of the above we have that the transformation

(4) $h := h\ \underline{m}\ [p\ q]$

is accomplished by H: swap. (p, q) , independent of the question whether p and q occur in the same ring of h or in different rings of h .

From R3, the one-pivot rule, we conclude that

(5) $h := h \underline{m} [p\ q\ r]$

is the same as

$h := h \underline{m} [p\ q] \underline{m} [q\ r]$,

which, by (4), is accomplished by "H: swap. (p, q); H: swap. (q, r)".

Development of the algorithm

Now we can specify the algorithm square more precisely. For a ring h and an array H coupled by the convention (1) we want to construct a program square satisfying

(0) $\{h = [U]\}$ square $\{h = [U] \underline{m} [U]\}$:

the ultimate program text has to be expressed in terms of H, without the use of auxiliary arrays.

The development consists of three steps. In step (i) we are concerned with the correctness of the final answer. In step (ii) we focus on the desired "in-situity" of the algorithm. In step (iii) we perform a coordinate transformation so as to express the ultimate text in terms of H.

(i) Having established the possibility to mingle with small rings, we propose the introduction of an invariant P defined by

P: $h \underline{m} [pX] = [U] \underline{m} [U]$,

which our program shall maintain while shrinking X. For reasons explained later, X shrinks by two elements at a time. This leads to the first version

of our algorithm.

```
{ h = [U] }
p, X : pX = U {P, see Note below}
; do length of X ≥ 2
  → q, r, Y : qrY = X
      {hence we have in combination with P:
       h m [p q r Y] = [U] m [U], or
       - using the one-pivot rule from
        ring calculus, pivot r -
       h m [p q r] m [r Y] = [U] m [U],
       so that the statements }
    ; h := h m [p q r]
    ; p, X := r, Y
        {reestablish P. And, indeed, X
         has shrunk by two elements.}
  od
  {P ∧ Length of X < 2}
; h := h m [p X]
  {h = [U] m [U]}.
```

Note A statement like $p, X : pX = U$ is supposed to establish that p and X satisfy $pX = U$. (End of Note.)

So much for the invariance of P on which the correctness of the final answer is based.

(ii) In the above algorithm, h is massaged towards its final value "under control of" the elements of sequence pX. For the algorithm to be in-situ, no additional storage should be reserved for these elements. The only possibility left is to identify them with elements of h. The simplest way to allocate

the elements of pX in h is to choose pX to be a contiguous segment in one of the rings of h. Since this, however, is not easily maintained -which we do not demonstrate here- we choose the next simples ways, viz. all elements but one contiguous in a ring of h. This would lead to an invariant

[p F X G] is a ring of h.

In fact, as follows from the program below, we can maintain the stronger

Q: h = [p F X G]

The next version of our algorithm then becomes

```
{h = [U]}
p, X: pX = U ; F, G := empty, empty   {Q}
; do length of X ≥ 2
→ q, r, Y: qrY = X
      {hence we have in combination with Q:
       h = [p F q r Y G], so that the
       statement}
  ; h := h m [p q r]
      {establishes h = [pFqrYG] m [pqr], or
       -using rule R8 from ring calculus -
       h = [rFqYGp] . Hence, }
  ; F, G := Fq, Gp
      {establishes h = [rFYG] and hence}
  ; p, X := r, Y
      {reestablishes Q}
od
{Q ∧ length of X < 2}
; h := h m [pX] .
```

Note The first and the second version of our algorithm only differ in that the second version contains assignments to the newly introduced variables F and G. Hence, the invariance of P is still guaranteed.
(End of Note.)

Remark An immediate consequence of our decision to shrink X by two elements at a time is that in the repetition h is mingled with a ring of length 3. Rules R3 and R6 from ring calculus show that h would not necessarily have remained a ring had we chosen to shrink X by one element at a time.
(End of Remark.)

So much for the invariance of Q which caters for the desired "in-situity" of the algorithm.

(iii) Our remaining obligation is to express the above algorithm in terms of the array H related to h by the representational convention (1).

- The guard "length of $X \geq 1$".
  Guided by Q, $h = [p\ F \times G]$, we introduce yet another invariant, QP, defined by

  QP: c = the first element of XGp $\wedge$
  d = the first element of Gp.

  We then have

  $$\text{length of } X = 0 \quad \equiv \quad c = d$$
  $$\text{length of } X = 1 \quad \equiv \quad c \neq d \wedge H.c = d$$

  Hence, the guard is

  $$c \neq d \wedge H.c \neq d.$$

For the invariance of QP we reformulate its first conjunct, using Q, as

$$c = H.(\text{the last element of } pF)$$

The invariance of QP is realized by, firstly, extending the initialization with $c, d := H.p, p$, and, secondly, inserting the statement $c := H.q$ after the assignments "$F, G := Fq, Gp$; $p, X := r, Y$".

- The statement $q, r, Y : qrY = X$.
  From QP, the guard and Q we derive $q = c$ and $r = H.q$ should hold, which is established by $q, r := c, H.c$.

- Concerning the statement $h := h\ \underline{m}\ [pX]$, with precondition length of $X < 2$, we note that for length of $X = 1$, $X = c$ holds (see QP) and that, hence, the statement is $h := h\ \underline{m}\ [p\,c]$.

Assembling the above parts and leaving out all references to variables h, U, X, F, G, and Y, which have become thought variables, and using the transformations (2), (4), and (5), we arrive at our ultimate program

```
{p = any element of the ring to be squared}
c, d := H.p, p
; do c ≠ d ∧ H.c ≠ d
  → q, r := c, H.c ; H: swap.(p,q) ; H: swap.(q,r)
    ; p := r ; c := H.q
  od
; if c = d → skip [] c ≠ d ∧ H.c = d → H: swap.(p,c) fi
```

Eindhoven,
17 September 1905

A.J.M. van Gasteren
W.H.J. Feijen