Two exercises in functional programming

0. Introduction

The following two programming problems were posed to me by Martin Rem. They seem to be en vogue among people who are interested in designing, so-called, systolic arrays. Here, I shall present derivations of functional programs for these problems. These derivations are completely standard and the programs thus obtained are surprisingly simple. Therefore, I am tempted to consider these exercises as a confirmation of the effectiveness of the techniques employed.

1. Notational conventions

In what follows, lower case letters x and z denote "elements", i.e. values of unspecified type, whereas upper case letters X, Y, and Z denote finite sequences of elements. Furthermore, for infinite sequence s and natural n the initial segment of length n of s will be denoted by $s{\downarrow}n$, i.e: $s{\downarrow}n$ is a finite sequence of length n and $(\underline{A}i: 0 \le i < n: (s{\downarrow}n).i = s.i)$.

2. Palindromes

"Being a palindrome" is a property of finite sequences; it is defined recursively as follows:

P0: each sequence of length at most 1 is a palindrome, P1 : a sequence of length at least 2 is a palindrome if it is of the form xXx, where X is a palindrome.

We now consider the function p, mapping infinite sequences to infinite boolean sequences, according to the specification:

$p \cdot s \cdot i \;=\; s{\downarrow}i$ is a palindrome , $i \geq 0$.

The problem is to derive a functional program for p. As usual we shall do so by mathematical induction on i, taking into account the structure of P0 and P1.
Firstly, for $i : 0 \leq i < 2$ we derive:

$s{\downarrow}i$ is a palindrome
= { $s{\downarrow}i$ has length at most 1 , P0 }
true
= { definition of : (twice) , <u>note 0</u> }
$(\text{true} : \text{true} : \; ? \;) \cdot i$.

(<u>note 0</u> : the question mark indicates that we have not yet decided upon the tail of the sequence, it being irrelevant at this stage, and that we do not even wish to give it a name.)

Secondly, for $i : i \geq 0$ we derive:

$(a{:}s){\downarrow}(i{+}2)$ is a palindrome
= { $(a{:}s){\downarrow}(i{+}2)$ has length at least 2 , P1 }
$(a{:}s)\cdot 0 = (a{:}s)\cdot(i{+}1) \;\wedge\; s{\downarrow}i$ is a palindrome
= { definition of : (twice), induction hypothesis }
$a = s\cdot i \;\wedge\; p\cdot s\cdot i$
= { introduction of a function q , see below }
$q\cdot a\cdot s\cdot(p\cdot s)\cdot i$

= { definition of : and the ? trick (twice) }
$(\,?\,:\,?\,:\, q.a.s.(p.s)).(i+2)$.

Due to the question marks the two results fit nicely together, such that abstraction from i yields:

$$p.(a:s) \;=\; \mathit{true} : \mathit{true} : q.a.s.(p.s) \;.$$

For elements a the function $q.a$ maps an (infinite) sequence and a boolean sequence to a boolean sequence according to the specification:

$$q.a.s.t.i \;=\; a=s.i \wedge t.i \quad , \; i \geq 0 \,.$$

This is such a simple specification that we leave the derivation of the following program to the interested reader:

$$q.a.(b:s).(c:t) \;=\; (a=b \wedge c) : q.a.s.t \;.$$

3. Carrés

"Being a carré" is a property of finite sequences; it is defined as follows:

C0: a finite sequence is a carré if it is of the form XX.

We now consider the function f, mapping (infinite) sequences to boolean sequences, according to the specification:

$$f.s.i \;=\; s{\downarrow}(2*i) \text{ is a carré} \;, \; i \geq 0 \,.$$

The problem is to derive a functional program for f.
The most marked difference between this exercise and the previous one is that definition C0 is not recursive. A recursive definition, however, might better suit our needs. Such a definition is not obvious: although a sequence of the form $xXxX$ is a carré, XxX is not a carré, due to the x "in the middle". This observation does, however, suggest a generalisation of carrés that might suffice. We call them k-carrés, for natural values k. The original carré then is a 0-carré:

G0: a finite sequence is a k-carré if it is of the form XYX, where Y has length k.

Without proof we state that k-carrés have the following properties:

G1: each sequence of length k is a k-carré,
G2: for sequences X, Y, Z, satisfying: X and Z have equal lengths and: Y has length k, we have:
$xXYzZ$ is a k-carré $=$
$x = z \;\wedge\; XYzZ$ is a $(k+1)$-carré .

Apart from this, no further inventions are required. The following derivation is, although somewhat complicated by the extra parameter k, quite similar to the previous one.
Firstly, we observe that the generalisation introduced above calls for a similar generalisation of the function f. I.e. we write:

$$f \cdot s \;=\; g \cdot 0 \cdot s \,,$$

where the function g has specification:

$g.k.s.i \;=\; s\downarrow(2*i+k)$ is a k-carré , $i \geq 0$.

Secondly, we derive:

$s\downarrow k$ is a k-carré
= { $s\downarrow k$ has length k, G1 }
true
= { definition of : and the ? trick }
(true : ?) . 0 .

Thirdly, for $i : i \geq 0$ we derive:

$(a:s)\downarrow(2*(i+1)+k)$ is a k-carré
= { $(a:s)\downarrow(2*(i+1)+k)$ has length $2*(i+1)+k$, G2 }
$(a:s).0 = (a:s).(i+1+k) \;\wedge\; s\downarrow(2*i+k+1)$ is a $(k+1)$-carré
= { definition of : (twice), induction hypothesis }
$a = s.(i+k) \;\wedge\; g.(k+1).s.i$
= { tl^k denotes k-fold application of tl }
$a = tl^k.s.i \;\wedge\; g.(k+1).s.i$
= { using the same — coincidence! — function q as before }
$q.a.(tl^k.s).(g.(k+1).s).i$
= { definition of : and the ? trick }
$(\;?\; : \; q.a.(tl^k.s).(g.(k+1).s)).(i+1)$.

Combination of these results gives the following program:

```
f.s         = g.0.s   where
g.k.(a:s)   =  true : q.a.(tl^k.s).(g.(k+1).s)   end ,
```

where the function q isthe same one as in the previous section.

As a final optimisation, an additional parameter can be introduced to represent the value of $tl^k.s$, thereby eliminating the expression itself from the program. After this — straightforward — program transformation the parameter k has become superfluous; hence, it can be eliminated. This results in the following program in which the transformed version of g has been named h (here, all relevant definitions have been put together):

```
f·s = h·s·s
      where h·(a:s)·(b:t) = true : q·a·t·(h·s·(tl·t))
          ; q·a·(b:s)·(c:t) = (a=b ∧ c) : q·a·s·t
      end .
```

The specification of h now is:

$$t = tl^k.s \Rightarrow (h.s.t.i = s\downarrow(2*i+k) \text{ is a k-carré}) ,$$

for all s, t, k, and i .

(<u>remark for formalists</u>: the degree of formality used in the above definitions of palindrome and carré has been deliberately chosen so as to suit my purpose.)

1986.12.10
Rob Hoogerwoord
department of mathematics and computing science
Eindhoven University of Technology