A simple theorem and its applications

The following theorem has been invented and proved by JME Bos — one of our students — .

Theorem 0: functions f, g, F and values X satisfying

$$
\begin{array}{lll}
(0) & f.[] & = X \\
(1) & f.(b:s) & = F.b.(f.s) \\
(2) & g.x.[] & = x \\
(3) & g.x.(b:s) & = g.(F.b.x).s \ ,
\end{array}
$$

also satisfy: $f \circ rev = g.X$.

The definition of g being tail-recursive we shall prove this theorem using the "invariance theorem for tail-recursive definitions" introduced in RH81 [2]. The proof thus obtained is shorter than the original proof by Bos in which, however, the invariance theorem is not used.

proof: We prove $f \circ rev = g.X$ by proving $f.(rev.S) = g.X.S$ for all S. For fixed S, the invariance theorem allows us to conclude $g.X.S = f.(rev.S)$, provided that, for suitably chosen predicate P and natural function vf, the following proof obligations are met:

$$
\begin{array}{lll}
(4) & P.X.S & \\
(5) & (\underline{A}x :: & P.x.[] \Rightarrow x = f.(rev.S)) \\
(6) & (\underline{A}x,b,s :: & P.x.(b:s) \Rightarrow P.(F.b.x).s) \\
(7) & (\underline{A}x,b,s :: & P.x.(b:s) \Rightarrow vf.(F.b.x).s < vf.x.(b:s)) \ .
\end{array}
$$

The only proof obligation with respect to vf is (7). From (7), it easily follows that, independently of P, $vf.x.s = length.s$ is a good proposal. So much for vf and (7).

The choice of a suitable P requires more care. By inspection of (2) and (3) and the observation that we are interested in $g.X.S$ we conclude that $S = t +\!\!+ s$, for some t, probably will be part of P. In order to formulate this we might introduce t as an auxiliary parameter into the game. As we shall see later, the argument turns out to run more smoothly if we write $rev.S = rev.s +\!\!+ t$ instead (i.e. replace t by $rev.t$):

$$P.t.x.s \equiv rev.S = rev.s +\!\!+ t \wedge x = f.t .$$

With this P we derive:

(4): $P.t.X.S$
$= \{$ definition of $P \}$
$rev.S = rev.S +\!\!+ t \wedge X = f.t$
$= \{$ "choose" $t = [] \}\{$ definition of $+\!\!+ \}$
$true \wedge X = f.[]$
$= \{$ (0) $\}\{$ calculus $\}$
$true.$

So, $P.t.X.S$ is satisfyable, viz. for $t = []$.

(5): $P.t.x.[]$
$= \{$ definition of $P \}$
$rev.S = rev.[] +\!\!+ t \wedge x = f.t$
$= \{$ definition of rev and $+\!\!+ \}$
$rev.S = t \wedge x = f.t$
$\Rightarrow \{$ equals for equals $\}$
$x = f.(rev.S)$.

(6): $P.t.x.(b:s)$

$= \{$ definition of $P \}$

$rev.S = rev.(b:s) +\!\!+ t \wedge x = f.t$

$= \{$ definition of $rev \}\{$: and $+\!\!+$ calculus $\}$

$rev.S = rev.s +\!\!+ (b:t) \wedge x = f.t$

$\Rightarrow \{$ functional application, to replace x by $F.b.x \}$

$rev.S = rev.s +\!\!+ (b:t) \wedge F.b.x = F.b.(f.t)$

$= \{ (1) \}$

$rev.S = rev.s +\!\!+ (b:t) \wedge F.b.x = f.(b:t)$

$= \{$ definition of $P \}$

$P.(b:t).(F.b.x).s$.

The last two derivations contain one step with an appeal to the definition of rev each. In this respect these derivations are minimal. This is the justification why P has been chosen in this way. Notice that these derivations are of the type that hardly leave you any choices.

(end of theorem 0).

Example 0: The theorem gives us a program for rev for free. Let f be given by: $f.[] = []$ and $f.(b:s) = b:f.s$. Then f is the identity function, hence $f \circ rev = rev$. By application of theorem 0 we conclude that $rev = g.[]$, where

$$g.x.[] = x$$
$$g.x.(b:s) = g.(b:x).s \ .$$

(end of example).

Notice that the theorem is constructive in the sense that for given f satisfying (0) and (1) the definition of a

function g such that $f \circ rev = g \cdot X$ follows immediately from (2) and (3). So, the theorem can be used for program transformations. Example 0 is an indication that efficient programs may be thus obtainable.

Example 1: The "minimal segment-sum", ms say, of an integer sequence can be defined recursively in terms of itself and the "minimal initial-segment-sum", mis say, of that sequence:

$$
\begin{array}{lll}
(8) & ms.[\,] & = 0 \\
(9) & ms.(b:s) & = mis.(b:s) \ \underline{min} \ ms.s \\
(10) & mis.[\,] & = 0 \\
(11) & mis.(b:s) & = (b + mis.s) \ \underline{min} \ 0 \ .
\end{array}
$$

It is easy to convince oneself that $(\underline{A}s :: ms.s \leq 0)$. Using this and (11), definition (9) may be replaced by the equivalent:

$$
(9a) \quad ms.(b:s) = (b + mis.s) \ \underline{min} \ ms.s \ .
$$

Now, the pair (ms, mis) of functions may be considered as one composite function satisfying the requirements of f in theorem 0. Application of the theorem and some cleaning up — after all, we only are interested in one half of the function — yields a function g with:

$$
\begin{array}{lll}
g.x.y.[\,] & = & x \\
g.x.y.(b:s) & = & g.((b+y) \ \underline{min} \ x).((b+y) \ \underline{min} \ 0).s \ .
\end{array}
$$

Then: $ms \circ rev = g.0.0$; hence, because $ms \circ rev = ms$, $ms = g.0.0$.

The time complexity of the function $g\cdot 0\cdot 0$ is linear (in the length of its argument) whereas ms has quadratic time complexity.
(end of example).

Theorem 0 may be easily generalised into the following

Theorem 1: functions f, g, F and values X satisfying

$$\begin{aligned} f\cdot[] &= X \\ f\cdot(b:s) &= F\cdot b\cdot s\cdot(f\cdot s) \\ g\cdot t\cdot x\cdot[] &= x \\ g\cdot t\cdot x\cdot(b:s) &= g\cdot(b:t)\cdot(F\cdot b\cdot t\cdot x)\cdot s \end{aligned}$$

also satisfy: $f \circ rev = g\cdot[]\cdot X$.

proof: We construct functions $\mathcal{f}$, $\mathcal{g}$, and $\mathcal{F}$, the values of which are lists of length 2 (representing pairs), as follows:

$$\begin{aligned} \mathcal{F}\cdot b\cdot[x,s] &= [\ F\cdot b\cdot s\cdot x\ ,\ b:s\ ] \\ \mathcal{f}\cdot[] &= [\ X\ ,\ []\ ] \\ \mathcal{f}\cdot(b:s) &= \mathcal{F}\cdot b\cdot(\mathcal{f}\cdot s) \\ \mathcal{g}\cdot[x,t]\cdot[] &= [x,t] \\ \mathcal{g}\cdot[x,t]\cdot(b:s) &= \mathcal{g}\cdot(\mathcal{F}\cdot b\cdot[x,t])\cdot s\ . \end{aligned}$$

Using mathematical induction we can derive

$$\begin{aligned} &(12) \quad \mathcal{f}\cdot s = [\ f\cdot s\ ,\ s\ ] \\ &(13) \quad \mathcal{g}\cdot[x,t]\cdot s = [\ g\cdot t\cdot x\cdot s\ ,\ rev\cdot s +\!\!+ t\ ]\ . \end{aligned}$$

Using this and Theorem 0 we now can derive:

$$
\begin{array}{ll}
 & (f \circ rev).s \\
= & \{ (12) \} \\
 & (f \circ rev).s.0 \\
= & \{ \text{Theorem 0, with } f,g,F,X := f,g,\mathcal{F},[X,[]] \} \\
 & g.[X,[]].s.0 \\
= & \{ (13) \} \\
 & g.[].X.s, \qquad \text{which does the job.}
\end{array}
$$

(end of Theorem 1).

Theorem 0 is nice because of its simplicity. The emergence of the function rev is somewhat surprising and might, for some applications, be annoying. It would, therefore, be nice if we could characterise the circumstances under which the function f in Theorem 0 satisfies: $f \circ rev = f$. I.e. which additional condition must be imposed on F such that $f \circ rev = f$? To analyse this situation we try to <u>prove</u> $f \circ rev = f$ in the hope to discover what is needed to complete the proof.

Firstly, we observe that $rev.s = s$ for all sequences s of length at most 1. Therefore, we confine our attention to sequences of at least 2 elements and we explicitly name the first and the last elements of such sequences.

Secondly, we derive:

$$
\begin{array}{ll}
 & f.(rev.([b] +\!\!+ s +\!\!+ [c])) \\
= & \{ \quad \text{properties of rev and } +\!\!+ \} \\
 & f.(c : rev.([b] +\!\!+ s)) \\
= & \{ (1) \} \\
 & F.c.(f.rev.([b] +\!\!+ s)) \\
= & \{ \text{induction hypothesis} \}
\end{array}
$$

$$
\begin{array}{ll}
 & F.c.(f.([b] +\!\!+ s)) \\
= & \{ \ (1) \ \} \\
 & F.c.(F.b.(f.s)) \ .
\end{array}
$$

Thirdly, we derive:

$$
\begin{array}{ll}
 & f.([b] +\!\!+ s +\!\!+ [c]) \\
= & \{ \ (1) \ \} \\
 & F.b.(f.(s +\!\!+ [c])) \\
= & \{ \text{ induction hypothesis } \} \\
 & F.b.(f.(rev.(s +\!\!+ [c]))) \\
= & \{ \text{ properties of } rev \text{ and } +\!\!+ \ \} \\
 & F.b.(f.(\ c : rev.s)) \\
= & \{ \ (1) \ \} \\
 & F.b.(F.c.(f.(rev.s))) \\
= & \{ \text{ induction hypothesis } \} \\
 & F.b.(F.c.(f.s)) \ .
\end{array}
$$

From these derivations we observe that a sufficient condition to conclude equality of $forev$ and f is

(14) $\quad F.b.(F.c.x) \ = \ F.c.(F.b.x)$, for all b, c, x.

For functions F satisfying (14), Theorem 0 turns out to be a very special case of D.C. Cooper's theorem [0], [1]. If we write F as a binary infix operator we get

$$b \ F \ (c \ F \ x) \ = \ c \ F \ (b \ F \ x) \text{ , for all } b, c, x .$$

Notice that F is of type $U \times V \to V$, for some U and V. As a special case, if F is of type $U \times U \to U$ and if F is symmetric and associative then F satisfies (14).

references

[0] D.C. Cooper, "The equivalence of certain computations", Computer Journal 9, 1 (1966).

[1] C. Hemerik, "Cooper's theorem", Eindhoven University of Technology, internal memorandum CH26 (1985).

[2] Rob Hoogerwoord, "Een invariantiestelling voor staartrecursieve functiedefinities", Eindhoven University of Technology, internal memorandum RH81 (1985).

1986.12.19
Rob Hoogerwoord
department of mathematics and computing science
Eindhoven University of Technology.