

Calculating with pointers

Introduction

This note presents a calculational method for dealing with pointers in weakest precondition semantics. It aims at facilitating the verification of program fragments that use pointers, without recourse to operational reasoning.

It is true that the unrestricted use of pointers may be considered somewhat old-fashioned. There is a growing consensus (to which the present author subscribes) that the derivation of correct programs is much to be preferred over any a posteriori verification. Types such as lists and trees have mathematical properties that are simpler than those of pointers, hence are more useful in program derivation. Recent publications concerning pointers typically propose schemes for their abolition [6, 7]. Nevertheless the study of pointers retains some importance, because the more abstract types are usually implemented by means of pointers and one wishes to prove the correctness of such an implementation. Moreover, there exist algorithms that exploit the aliasing effect provided by pointers for efficiency reasons [3].

Among existing approaches to the problem at hand, the one proposed by Morris [4, 5] bears the closest resemblance to the method outlined here; the main differences are the purely calculational method of our method and its applicability to arbitrary postconditions rather than just conditions expressing the reachability of certain nodes.

Throughout, we limit ourselves to Pascal-like pointers. These are more restricted than the pointers in some other languages, since

- (i) pointers of the same type cannot refer to objects of different type,
- (ii) pointers of different types cannot have the same value, except for nil,
- (iii) pointers may only refer to variables that have no explicit name declared in the program [8].

Following [2], we associate with every pointer type $P = \hat{T}$ a thought variable m of type array [P] of T such that for every p of type P the component $m[p]$ is identified with p^{\wedge} . Then (ii) and (iii) ensure that assignments to components of m influence no program variables and no other thought variables; hence they may be described by means of the usual array semantics [1, 2].

The consequences of this observation will be demonstrated in the following concrete case. Throughout this note, the type definitions and declarations

```

(0)  type   ptr = ^rec;
      rec = record
          c: Integer;
          s: ptr
      end;
      var u, v, w: ptr;

```

are considered given. We allow ourselves to deviate slightly from the syntax of Pascal in that nil^s will be considered a valid expression with the value nil . This decision is taken in order to avoid a considerable amount of case analysis in the proofs of the theorems.

The choice of the types ptr and rec was influenced by our desire to keep all calculations as simple as possible; our method can be applied in the case of records with more than one pointer field, but only at the price of more involved calculations [0].

By m we denote the thought array associated with ptr in the above sense. From [1] we adapt the following notation: for p, q of type ptr and r of type rec we denote by $(m; p: r)$ an array equal to m except that $m[p] = r$ and by $(m; p, s: q)$ an array equal to m except that $m[p].s = q$. More precisely,

```

(1)  (m; p: r)[i] =   if i = p → r
                   □ i ≠ p → m[i]
                   fi   ,

```

```

(2)  (m; p, s: q)[i].s = if i = p → q
                       □ i ≠ p → m[i].s
                       fi   ,

```

```

(3)  (m; p, s: q)[i].c = m[i].c   .

```

EXAMPLE 0 ([4], Example 10). Assume $u \neq \text{nil}$. Then

$$\text{wp}(u^s := v, w^s.s = v)$$

$$\begin{aligned}
&= \{\text{introduction of } m\} \\
&\quad \text{wp}(m[u].s := v, m[m[w].s].s = v) \\
&= \{\text{array semantics}\} \\
&\quad (m; u, s: v)[(m; u, s: v)[w].s].s = v \\
&= \{(2) \text{ with } p, q, i := u, v, w\} \\
&\quad \underline{\text{if}} \ u = w \rightarrow (m; u, s: v)[v].s = v \\
&\quad \square \ u \neq w \rightarrow (m; u, s: v)[m[w].s].s = v \\
&\quad \underline{\text{fi}} \\
&= \{(2) \text{ with } p, q, i := u, v, v \text{ and } p, q, i := u, v, m[w].s\} \\
&\quad \underline{\text{if}} \ u = w \rightarrow \underline{\text{if}} \ u = v \rightarrow \text{true} \\
&\quad \quad \square \ u \neq v \rightarrow m[v].s = v \\
&\quad \quad \underline{\text{fi}} \\
&\quad \square \ u \neq w \rightarrow \underline{\text{if}} \ u = m[w].s \rightarrow \text{true} \\
&\quad \quad \square \ u \neq m[w].s \rightarrow m[m[w].s].s = v \\
&\quad \underline{\text{fi}} \\
&= \{\text{elimination of } \underline{\text{if}} \text{ and } m\} \\
&\quad (u = w \wedge u = v) \\
&\quad \vee (u = w \wedge u \neq v \wedge v.s = v) \\
&\quad \vee (u \neq w \wedge u = w.s) \\
&\quad \vee (u \neq w \wedge u \neq w.s \wedge w.s.s = v) \quad .
\end{aligned}$$

Assignment to an object pointed to is equally easy to handle, as the next example shows.

EXAMPLE 1. Assume $u \neq \text{nil}$, $u.s \neq \text{nil}$, $v \neq \text{nil}$. Then

$$\begin{aligned}
&\text{wp}(u.s := v, w.c = 1) \\
&= \{\text{introduction of } m\} \\
&\quad \text{wp}(m[m[u].s] := m[v], m[w].c = 1) \\
&= \{\text{array semantics}\} \\
&\quad (m; m[u].s: m[v])[w].c = 1 \\
&= \{(1) \text{ with } p, r, i := m[u].s, m[v], w\} \\
&\quad \underline{\text{if}} \ m[u].s = w \rightarrow m[v].c = 1 \\
&\quad \square \ m[u].s \neq w \rightarrow m[w].c = 1
\end{aligned}$$

$$\begin{aligned} & \underline{fi} \\ = & \{ \text{elimination of } \underline{if} \text{ and } m \} \\ & (u^{\wedge}.s = w \wedge v^{\wedge}.c = 1) \vee (u^{\wedge}.s \neq w \wedge w^{\wedge}.c = 1) \quad . \end{aligned}$$

The preceding examples are all rather trivial in the sense that the number of pointers involved is no more than 4. Much more difficult problems arise when the number is not explicitly limited. In order to be able to state and solve such problems, we introduce the following notation. Let p and q denote arbitrary values of type ptr and i an arbitrary natural number (a convention we shall silently use in the rest of this note). Put

$$(4) \quad \alpha_m(i, p) = \begin{aligned} & \underline{if} \ i = 0 \rightarrow p \\ & \square \ i > 0 \rightarrow m[\alpha_m(i-1, p)].s \\ & \underline{fi} \quad , \end{aligned}$$

$$(5) \quad \beta_m(p, q) = (\underline{MIN} \ i: i \geq 1 \wedge \alpha_m(i-1, p) = q: i) \quad .$$

Informally, $\alpha_m(i, p)$ is the pointer obtained by postfixing p a total of i times with $\wedge.s$ and $\beta_m(p, q)$ is the length of the sequence $p, p^{\wedge}.s, p^{\wedge}.s^{\wedge}.s, \dots, q$. In the interest of brevity, we shall write α and β instead of α_m and β_m and trust the reader to remember that their value depends on m . One type of problem we will consider is concerned with reachability: a representative question would be to determine

$$wp(p^{\wedge}.s := q, \beta(u, v) < \infty) \quad .$$

In [5], problems like this one are treated by means of graph theory. We shall solve this problem in Example 2. Another type concerns sequences of values: we introduce the notation

$$(6) \quad \text{seq}(p) = (\underline{SEQ} \ i: 0 \leq i < \beta(p, \text{nil}) - 1: m[\alpha(i, p)].c) \quad ,$$

i.e., the list of integers pointed to by p , and we would like to compute

$$wp(p^{\wedge}.s := q, \text{seq}(u) = S)$$

for a given sequence S of integers. This problem will be solved in Example 3. As a final

demonstration, the theorems will be applied to prove correctness of a list insertion algorithm.

Theorems and proofs

We now proceed to study the behaviour of α and β under assignments to m . An expression followed by $(x := e)$ denotes that expression with all free occurrences of x replaced by e .

LEMMA 0. If, for some natural j and k ,

$$(7) \quad \alpha(j, p) = \alpha(j + k, p) \quad ,$$

then, for every natural i ,

$$\alpha(j + i, p) = \alpha(j + i \bmod k, p) \quad .$$

PROOF. Induction on i . The base, $i = 0$, is trivial. In order to prove the induction step, we remark that, for $i > 0$,

$$\begin{aligned} & \alpha(j + i, p) \\ &= \{(4), j + i > 0\} \\ & \quad m[\alpha(j + i - 1, p)].s \\ &= \{\text{induction hypothesis}\} \\ & \quad m[\alpha(j + (i - 1) \bmod k, p)].s \\ &= \{(4)\} \\ & \quad \alpha(j + (i - 1) \bmod k + 1, p) \\ &= \{\text{definition of } \bmod\} \\ & \quad \underline{\text{if}} \ i \bmod k > 0 \rightarrow \alpha(j + i \bmod k, p) \\ & \quad \square \ i \bmod k = 0 \rightarrow \alpha(j + k, p) \\ & \quad \underline{\text{fi}} \\ &= \{(7)\} \\ & \quad \underline{\text{if}} \ i \bmod k > 0 \rightarrow \alpha(j + i \bmod k, p) \\ & \quad \square \ i \bmod k = 0 \rightarrow \alpha(j, p) \end{aligned}$$

$$\begin{aligned}
& \underline{fi} \\
= & \{ \} \\
& \alpha(j + i \bmod k, p) \quad .
\end{aligned}$$

THEOREM 0.

$$\begin{aligned}
& \alpha(i, u)(m := (m; p, s: q)) \\
= & \underline{if} \ 0 \leq i < \beta(u, p) \rightarrow \alpha(i, u) \\
& \square \ i \geq \beta(u, p) \quad \rightarrow \alpha((i - \beta(u, p)) \bmod \beta(q, p), q) \\
& \underline{fi} \quad .
\end{aligned}$$

PROOF. For $0 \leq i \leq \beta(u, p) + \beta(q, p)$, we prove the statement by induction on i . The base, $i = 0$, uses the fact that u is a variable, not an expression involving m . For the step, one observes that, for $0 < i \leq \beta(u, p) + \beta(q, p)$,

$$\begin{aligned}
& \alpha(i, u)(m := (m; p, s: q)) \\
= & \{(4), i > 0\} \\
& (m[\alpha(i-1, u)].s)(m := (m; p, s: q)) \\
= & \{\text{distributing the substitution}\} \\
& (m; p, s: q)[\alpha(i-1, u)(m := (m; p, s: q)].s \\
= & \{\text{induction hypothesis}\} \\
& \underline{if} \ 0 < i \leq \beta(u, p) \rightarrow (m; p, s: q)[\alpha(i-1, u)].s \\
& \square \ \beta(u, p) < i \leq \beta(u, p) + \beta(q, p) \\
& \quad \rightarrow (m; p, s: q)[\alpha(i-1 - \beta(u, p), q)].s \\
& \underline{fi} \\
= & \{(2) \text{ with } i := \alpha(i-1, u) \text{ and with } i := \alpha(i-1 - \beta(u, p), q)\} \\
& \underline{if} \ 0 < i \leq \beta(u, p) \\
& \quad \rightarrow \underline{if} \ \alpha(i-1, u) \neq p \rightarrow m[\alpha(i-1, u)].s \\
& \quad \square \ \alpha(i-1, u) = p \rightarrow q \\
& \quad \underline{fi} \\
& \square \ \beta(u, p) < i \leq \beta(u, p) + \beta(q, p) \\
& \quad \rightarrow \underline{if} \ \alpha(i-1 - \beta(u, p), q) \neq p \rightarrow m[\alpha(i-1 - \beta(u, p), q)].s \\
& \quad \square \ \alpha(i-1 - \beta(u, p), q) = p \rightarrow q \\
& \quad \underline{fi}
\end{aligned}$$

$$\begin{aligned}
& \underline{f_i} \\
= & \{(5) \text{ with } p, q := u, p \text{ and with } p, q := q, p\} \\
& \underline{\text{if}} \ 0 < i < \beta(u, p) \rightarrow m[\alpha(i-1, u)].s \\
& \square \ i = \beta(u, p) \rightarrow q \\
& \square \ \beta(u, p) < i < \beta(u, p) + \beta(q, p) \rightarrow m[\alpha(i-1 - \beta(u, p), q)].s \\
& \square \ i = \beta(u, p) + \beta(q, p) \rightarrow q \\
& \underline{f_i} \\
= & \{(4)\} \\
& \underline{\text{if}} \ 0 < i < \beta(u, p) \rightarrow \alpha(i, u) \\
& \square \ \beta(u, p) \leq i < \beta(u, p) + \beta(q, p) \rightarrow \alpha(i - \beta(u, p), q) \\
& \square \ i = \beta(u, p) + \beta(q, p) \rightarrow \alpha(0, q) \\
& \underline{f_i} \ .
\end{aligned}$$

This proves the theorem for $0 \leq i \leq \beta(u, p) + \beta(q, p)$.

In particular, comparing the results obtained for $i = \beta(u, p)$ and for $i = \beta(u, p) + \beta(q, p)$ gives

$$(8) \quad \alpha(\beta(u, p), u) (m := m') = \alpha(\beta(u, p) + \beta(q, p), u) (m := m') \quad ,$$

where m' is short for $(m; p, s: q)$. We now observe

$$\begin{aligned}
& (8) \\
> & \{\text{Lemma 0 with } j, k, m := \beta(u, p), \beta(q, p), m'\} \\
& (\underline{A} \ i :: \alpha(\beta(u, p) + i, u) (m := m') \\
& \quad = \alpha(\beta(u, p) + i \underline{\text{mod}} \beta(q, p), u) (m := m') \\
&) \\
= & \{\text{dummy transformation, } i := i - \beta(u, p)\} \\
& (\underline{A} \ i: i \geq \beta(u, p): \\
& \quad \alpha(i, u) (m := m') \\
& \quad = \alpha(\beta(u, p) + (i - \beta(u, p)) \underline{\text{mod}} \beta(q, p), u) (m := m') \\
&) \\
= & \{\beta(u, p) \leq \beta(u, p) + (i - \beta(u, p)) \underline{\text{mod}} \beta(q, p) < \beta(u, p) + \beta(q, p)\} \\
& (\underline{A} \ i: i \geq \beta(u, p): \\
& \quad \alpha(i, u) (m := m') = \alpha(i - \beta(u, p)) \underline{\text{mod}} \beta(q, p), q)
\end{aligned}$$

) .

This proves the theorem for all remaining values of i .

LEMMA 1.

$$\begin{aligned}
 & (\underline{\text{MIN}} \ i: 1 \leq i \leq N \wedge \alpha(i-1, p) = q: i) \\
 = & \underline{\text{if}} \ \beta(p, q) \leq N \rightarrow \beta(p, q) \\
 & \square \ \beta(p, q) > N \rightarrow \infty \\
 & \underline{\text{fi}} \ .
 \end{aligned}$$

PROOF.

$$\begin{aligned}
 & (\underline{\text{MIN}} \ i: 1 \leq i \leq N \wedge \alpha(i-1, p) = q: i) \\
 = & \{\alpha(i-1, p) = q > i \geq \beta(p, q)\} \\
 & (\underline{\text{MIN}} \ i: \beta(p, q) \leq i \leq N \wedge \alpha(i-1, p) = q: i) \\
 = & \{\alpha(i-1, p) = q < i = \beta(p, q)\} \\
 & \underline{\text{if}} \ \beta(p, q) \leq N \rightarrow \beta(p, q) \\
 & \square \ \beta(p, q) > N \rightarrow \infty \\
 & \underline{\text{fi}} \ .
 \end{aligned}$$

THEOREM 1.

$$\begin{aligned}
 & \beta(u, v) (m := (m; p, s: q)) \\
 = & \underline{\text{if}} \ \beta(u, v) \leq \beta(u, p) \rightarrow \beta(u, v) \\
 & \square \ \beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \rightarrow \beta(u, p) + \beta(q, v) \\
 & \square \ \beta(u, v) > \beta(u, p) \wedge \beta(q, v) > \beta(q, p) \rightarrow \infty \\
 & \underline{\text{fi}} \ .
 \end{aligned}$$

PROOF.

$$\begin{aligned}
 & \beta(u, v) (m := (m; p, s: q)) \\
 = & \{(5)\}
 \end{aligned}$$

$$\begin{aligned}
& (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha(i-1, u) = v: i) (m:= (m; p, s: q)) \\
= & \{v \text{ is independent of } m\} \\
& (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha(i-1, u) (m:= (m; p, s: q)) = v: i) \\
= & \{\text{Theorem 0}\} \\
& (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(u, p) \wedge \alpha(i-1, u) = v: i) \\
& \underline{\text{min}} \ (\underline{\text{MIN}} \ i: i > \beta(u, p) \wedge \alpha((i-1 - \beta(u, p)) \underline{\text{mod}} \beta(q, p), q) = v: i) \\
= & \{\text{dummy transformation, } i:= i + \beta(u, p)\} \\
& (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(u, p) \wedge \alpha(i-1, u) = v: i) \\
& \underline{\text{min}} \ (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha((i-1) \underline{\text{mod}} \beta(q, p), q) = v: i + \beta(u, p)) \\
= & \{\text{periodicity of } \underline{\text{mod}}\} \\
& (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(u, p) \wedge \alpha(i-1, u) = v: i) \\
& \underline{\text{min}} \ (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(q, p) \wedge \alpha(i-1, q) = v: i + \beta(u, p)) \\
= & \{\text{Lemma 1 with } p, q, N:= u, v, \beta(u, p) \\
& \text{and with } p, q, N:= q, v, \beta(q, p) \\
& \} \\
& \underline{\text{if}} \ \beta(u, v) \leq \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \\
& \quad \rightarrow \beta(u, v) \underline{\text{min}} \ (\beta(q, v) + \beta(u, p)) \\
\Box \ & \beta(u, v) \leq \beta(u, p) \wedge \beta(q, v) > \beta(q, p) \rightarrow \beta(u, v) \underline{\text{min}} \ \infty \\
\Box \ & \beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \rightarrow \infty \underline{\text{min}} \ (\beta(q, v) + \beta(u, p)) \\
\Box \ & \beta(u, v) > \beta(u, p) \wedge \beta(q, v) > \beta(q, p) \rightarrow \infty \underline{\text{min}} \ \infty \\
& \underline{\text{fi}} \\
= & \{\} \\
& \underline{\text{if}} \ \beta(u, v) \leq \beta(u, p) \rightarrow \beta(u, v) \\
\Box \ & \beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \rightarrow \beta(q, v) + \beta(u, p) \\
\Box \ & \beta(u, v) > \beta(u, p) \wedge \beta(q, v) > \beta(q, p) \rightarrow \infty \\
& \underline{\text{fi}} \quad .
\end{aligned}$$

REMARK. Theorem 1 and its proof remain valid when v is replaced by the constant nil .

When applying Theorem 1 to sequences of values, we shall make use of the following lemmata.

LEMMA 2. For $q \neq \text{nil}$,

$$\beta(p, q) < \beta(p, \text{nil}) \equiv \beta(p, q) < \infty \quad .$$

PROOF. From (4) it is easily proved by induction that

$$(9) \quad (\underline{A} \ i: i \geq \beta(p, \text{nil}): \alpha(i-1, p) = \text{nil}) \quad .$$

Hence,

$$\begin{aligned}
& \beta(p, q) < \infty \\
& = \{(5)\} \\
& \quad (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha(i-1, p) = q: i) < \infty \\
& = \{\text{domain split}\} \\
& \quad (\underline{\text{MIN}} \ i: 1 \leq i < \beta(p, \text{nil}) \wedge \alpha(i-1, p) = q: i) \\
& \quad \underline{\text{min}} \ (\underline{\text{MIN}} \ i: i \geq \beta(p, \text{nil}) \wedge \alpha(i-1, p) = q: i) < \infty \\
& = \{(9), q \neq \text{nil}\} \\
& \quad (\underline{\text{MIN}} \ i: 1 \leq i < \beta(p, \text{nil}) \wedge \alpha(i-1, p) = q: i) < \infty \\
& = \{(5)\} \\
& \quad \underline{\text{if}} \ \beta(p, \text{nil}) = \infty \rightarrow \beta(p, q) < \infty \\
& \quad \square \ \beta(p, \text{nil}) < \infty \\
& \quad \rightarrow (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(p, \text{nil}) - 1 \wedge \alpha(i-1, p) = q: i) < \infty \\
& \quad \underline{\text{fi}} \\
& = \{\text{Lemma 1 with } N := \beta(p, \text{nil}) - 1\} \\
& \quad \underline{\text{if}} \ \beta(p, \text{nil}) = \infty \rightarrow \beta(p, q) < \infty \\
& \quad \square \ \beta(p, \text{nil}) < \infty \\
& \quad \rightarrow \underline{\text{if}} \ \beta(p, q) \leq \beta(p, \text{nil}) - 1 \rightarrow \beta(p, q) < \infty \\
& \quad \quad \square \ \beta(p, q) > \beta(p, \text{nil}) - 1 \rightarrow \text{false} \\
& \quad \underline{\text{fi}} \\
& \quad \underline{\text{fi}} \\
& = \{\} \\
& \quad \beta(p, q) < \beta(p, \text{nil}) \quad .
\end{aligned}$$

LEMMA 3. For all i ,

$$\alpha(i, m[p].s) = \alpha(i \bmod \beta(m[p].s, p), m[p].s) \quad .$$

PROOF. In case $\beta(m[p].s, p) = \infty$, Lemma 3 is obviously true. Now assume $\beta(m[p].s, p) < \infty$. For every positive N ,

$$\begin{aligned}
& N = \beta(m[p].s, p) \\
\Rightarrow & \{(5)\} \\
& \alpha(N-1, m[p].s) = p \\
\Rightarrow & \{(4)\} \\
& \alpha(N, m[p].s) = m[p].s \\
= & \{(4)\} \\
& \alpha(N, m[p].s) = \alpha(0, m[p].s) \\
\Rightarrow & \{\text{Lemma 0 with } j, k, p := 0, N, m[p].s\} \\
& (\underline{A} \ i: i \geq 0: \alpha(i, m[p].s) = \alpha(i \bmod N, m[p].s)) \quad .
\end{aligned}$$

LEMMA 4. If $p \neq \text{nil}$ and $\text{seq}(p^{\wedge}.s)$ is finite, then $\beta(m[p].s, p) = \infty$.

PROOF. For $p \neq \text{nil}$,

$$\begin{aligned}
& \text{seq}(p^{\wedge}.s) \text{ is finite} \\
= & \{(6)\} \\
& \beta(m[p].s, \text{nil}) < \infty \\
= & \{(5)\} \\
& (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha(i-1, m[p].s) = \text{nil}: i) < \infty \\
= & \{\text{Lemma 3 with } i := i-1\} \\
& (\underline{\text{MIN}} \ i: i \geq 1 \wedge \alpha((i-1) \bmod \beta(m[p].s, p), m[p].s) = \text{nil}: i) < \infty \\
= & \{\text{periodicity of } \bmod\} \\
& (\underline{\text{MIN}} \ i: 1 \leq i \leq \beta(m[p].s, p) \wedge \alpha(i-1, m[p].s) = \text{nil}: i) < \infty \\
\Rightarrow & \{\text{Lemma 1 with } N, p, q := \beta(m[p].s, p), m[p].s, \text{nil}\} \\
& \beta(m[p].s, \text{nil}) \leq \beta(m[p].s, p) \\
= & \{\text{Lemma 2 with } p, q := m[p].s, p\} \\
& \beta(m[p].s, p) = \infty .
\end{aligned}$$

Applications

EXAMPLE 2 ([5], Example 1). Assume $p \neq \text{nil}$. Then

$$\begin{aligned}
& \text{wp}(p \wedge s := q, \beta(u, v) < \infty) \\
= & \{\text{introduction of } m \text{ ; array semantics}\} \\
& \beta(u, v) (m := (m; p, s: q)) < \infty \\
= & \{\text{Theorem 1}\} \\
& \underline{\text{if}} \beta(u, v) \leq \beta(u, p) \rightarrow \beta(u, v) < \infty \\
& \square \beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \rightarrow \beta(q, v) + \beta(u, p) < \infty \\
& \square \beta(u, v) > \beta(u, p) \wedge \beta(q, v) > \beta(q, p) \rightarrow \text{false} \\
& \underline{\text{fi}} \\
= & \{ \} \\
& (\beta(u, v) \leq \beta(u, p) \wedge \beta(u, v) < \infty) \\
& \vee (\beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \wedge \beta(q, v) < \infty) \quad .
\end{aligned}$$

Although this is in itself a satisfactory answer, the calculation may be carried a bit further in order to establish that the result we have obtained is identical to the one in [5].

$$\begin{aligned}
& (\beta(u, v) \leq \beta(u, p) \wedge \beta(u, v) < \infty) \\
& \vee (\beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \wedge \beta(q, v) < \infty) \\
= & \{\text{adding a new second disjunct that is a strengthening of the first}\} \\
& (\beta(u, v) \leq \beta(u, p) \wedge \beta(u, v) < \infty) \\
& \vee (\beta(u, v) \leq \beta(u, p) \wedge \beta(u, p) < \infty \\
& \quad \wedge \beta(q, v) \leq \beta(q, p) \wedge \beta(q, v) < \infty) \\
& \quad) \\
& \vee (\beta(u, v) > \beta(u, p) \wedge \beta(q, v) \leq \beta(q, p) \wedge \beta(q, v) < \infty) \\
= & \{\beta(u, v) > \beta(u, p) > \beta(u, p) < \infty, \text{distributing}\} \\
& (\beta(u, v) \leq \beta(u, p) \wedge \beta(u, v) < \infty) \\
& \vee (\beta(u, p) < \infty \wedge \beta(q, v) \leq \beta(q, p) \wedge \beta(q, v) < \infty) \quad .
\end{aligned}$$

Now let us introduce, for any p, q, r the notation

$$(10) \quad p \rightarrow q \equiv \beta(p, q) < \infty \quad ,$$

$$(11) \quad p \rightarrow q \mid r \equiv \beta(p, q) < \infty \wedge \beta(p, q) \leq \beta(p, r) \quad .$$

Then the result obtained can be written as

$$\begin{aligned} & \text{wp}(p^{\wedge}.s := q, u - v) \\ & \equiv u \rightarrow v \mid p \vee (u \rightarrow p \wedge q \rightarrow v \mid p) \quad , \end{aligned}$$

which is the formula given in [5].

EXAMPLE 3. Assume $p \neq \text{nil}$. Let S be a finite sequence of integers. By $++$ the concatenation operator is denoted. Then

$$\begin{aligned} & \text{wp}(p^{\wedge}.s := q, \text{seq}(u) = S) \\ & = \{\text{introduction of } m \text{ ; array semantics}\} \\ & \quad \text{seq}(u)(m := (m; p, s: q)) = S \\ & = \{(6)\} \\ & \quad (\underline{\text{SEQ}} \ i: 0 \leq i < \beta(u, \text{nil})(m := (m; p, s: q)) - 1: \\ & \quad \quad (m[\alpha(i, u)].c)(m := (m; p, s: q)) \\ & \quad) = S \\ & = \{\text{Theorem 1 and the remark following it; (3)}\} \\ & \quad \underline{\text{if}} \ \beta(u, \text{nil}) \leq \beta(u, p) \\ & \quad \rightarrow (\underline{\text{SEQ}} \ i: 0 \leq i < \beta(u, \text{nil}) - 1: \\ & \quad \quad m[\alpha(i, u)(m := (m; p, s: q))].c \\ & \quad \quad) = S \\ & \quad \square \ \beta(u, \text{nil}) > \beta(u, p) \ \& \ \beta(q, \text{nil}) \leq \beta(q, p) \\ & \quad \rightarrow (\underline{\text{SEQ}} \ i: 0 \leq i < \beta(q, \text{nil}) + \beta(u, p) - 1: \\ & \quad \quad m[\alpha(i, u)(m := (m; p, s: q))].c \\ & \quad \quad) = S \\ & \quad \square \ \beta(u, \text{nil}) > \beta(u, p) \ \& \ \beta(q, \text{nil}) > \beta(q, p) \\ & \quad \rightarrow (\underline{\text{SEQ}} \ i: i \geq 0: m[\alpha(i, u)(m := (m; p, s: q))].c) = S \\ & \quad \underline{\text{fi}} \\ & = \{\text{Lemma 2, using } p \neq \text{nil} \text{ ; finiteness of } S\} \\ & \quad \underline{\text{if}} \ \beta(u, p) = \infty \\ & \quad \rightarrow (\underline{\text{SEQ}} \ i: 0 \leq i < \beta(u, \text{nil}) - 1: \end{aligned}$$

$$\begin{aligned}
& m[\alpha(i, u)(m := (m; p, s: q))].c \\
&) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) = \infty \\
& \rightarrow (\text{SEQ } i: 0 \leq i < \beta(q, \text{nil}) + \beta(u, p) - 1: \\
& \quad m[\alpha(i, u)(m := (m; p, s: q))].c \\
& \quad) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) < \infty \rightarrow \text{false} \\
& \underline{\text{fi}} \\
= & \{\text{Theorem 0}\} \\
& \underline{\text{if}} \beta(u, p) = \infty \\
& \quad \rightarrow (\text{SEQ } i: 0 \leq i < \beta(u, \text{nil}) - 1: m[\alpha(i, u)].c) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) = \infty \\
& \rightarrow (\text{SEQ } i: 0 \leq i < \beta(u, p): m[\alpha(i, u)].c) \\
& \quad ++ (\text{SEQ } i: \beta(u, p) \leq i < \beta(q, \text{nil}) + \beta(u, p) - 1: \\
& \quad \quad m[\alpha(i - \beta(u, p), q)].c \\
& \quad) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) < \infty \rightarrow \text{false} \\
& \underline{\text{fi}} \\
= & \{\text{dummy transformation, } i := i + \beta(u, p) ; (6)\} \\
& \underline{\text{if}} \beta(u, p) = \infty \rightarrow \text{seq}(u) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) = \infty \\
& \rightarrow (\text{SEQ } i: 0 \leq i < \beta(u, p): m[\alpha(i, u)].c) ++ \text{seq}(q) = S \\
\Box & \beta(u, p) < \infty \wedge \beta(q, p) < \infty \rightarrow \text{false} \\
& \underline{\text{fi}} \\
= & \{\} \\
& (\beta(u, p) = \infty \wedge \text{seq}(u) = S) \\
& \vee (\beta(u, p) < \infty \wedge \beta(q, p) = \infty \\
& \quad \wedge (\text{SEQ } i: 0 \leq i < \beta(u, p): m[\alpha(i, u)].c) ++ \text{seq}(q) = S \\
& \quad) .
\end{aligned}$$

If, in addition to (10), we introduce the notation

$$(12) \quad \text{seq}_p(u) = (\text{SEQ } i: 0 \leq i < \beta(u, p): m[\alpha(i, u)].c) \quad ,$$

the result obtained may be written as

$$\begin{aligned} & \text{wp}(p^{\wedge}.s := q, \text{seq}(u) = S) \\ & \equiv (\neg u \rightarrow p \wedge \text{seq}(u) = S) \\ & \quad \vee (u \rightarrow p \wedge \neg q \rightarrow p \ \& \ \text{seq}_p(u) \ ++ \ \text{seq}(q) = S) \quad . \end{aligned}$$

List_insertion

EXAMPLE 4. Let u point to a linked list of integers. Let v point to some node in the list; in other words, assume

$$(13) \quad v \neq \text{nil} \quad ,$$

$$(14) \quad \beta(u, v) < \infty \quad .$$

Let w point to a node that is not in the list; in other words, assume

$$(15) \quad w \neq \text{nil} \quad ,$$

$$(16) \quad \beta(u, w) = \infty \quad .$$

The usual way to insert node w^{\wedge} after v^{\wedge} in the list is to perform

$$w^{\wedge}.s := v^{\wedge}.s; v^{\wedge}.s := w \quad .$$

This trick is at the heart of many list-processing procedures. In order to show its correctness, we shall compute, for a given finite sequence S of integers,

$$\text{wp}(w^{\wedge}.s := v^{\wedge}.s; v^{\wedge}.s := w, \text{seq}(u) = S)$$

under assumption of (13) through (16).

Applying Example 3 with $p, q := v, w$ gives for $\text{wp}(v^{\wedge}.s := w, \text{seq}(u) = S)$ the predicate

$$\begin{aligned}
(17) \quad & (\beta(u, v) = \infty \wedge (\text{SEQ } i: 0 \leq i < \beta(u, \text{nil}): m[\alpha(i, u)].c) = S) \\
& \vee (\beta(u, v) < \infty \wedge \beta(w, v) = \infty \\
& \quad \wedge (\text{SEQ } i: 0 \leq i < \beta(u, v): m[\alpha(i, u)].c) \\
& \quad ++ (\text{SEQ } i: 0 \leq i < \beta(w, \text{nil}) - 1: m[\alpha(i, w)].c) = S \\
&) \quad .
\end{aligned}$$

We now proceed as follows.

$$\begin{aligned}
& \beta(u, v) (m := (m; w, s: m[v].s)) \\
= & \{ \text{Theorem 1 with } p, q := w, m[v].s \} \\
& \underline{\text{if}} \beta(u, v) \leq \beta(u, w) \rightarrow \beta(u, v) \\
& \square \beta(u, v) > \beta(u, w) \wedge \beta(m[v].s, v) \leq \beta(m[v].s, w) \\
& \quad \rightarrow \beta(u, w) + \beta(m[v].s, v) \\
& \square \beta(u, v) > \beta(u, w) \wedge \beta(m[v].s, v) > \beta(m[v].s, w) \rightarrow \infty \\
& \underline{\text{fi}} \\
= & \{ (16) \} \\
& \beta(u, v) \quad , \\
& \beta(w, v) (m := (m; w, s: m[v].s)) \\
= & \{ \text{Theorem 1 with } u, p, q := w, w, m[v].s \} \\
& \underline{\text{if}} \beta(w, v) \leq \beta(w, w) \rightarrow \beta(w, v) \\
& \square \beta(w, v) > \beta(w, w) \wedge \beta(m[v].s, v) \leq \beta(m[v].s, w) \\
& \quad \rightarrow \beta(w, w) + \beta(m[v].s, v) \\
& \square \beta(w, v) > \beta(w, w) \wedge \beta(m[v].s, v) > \beta(m[v].s, w) \rightarrow \infty \\
& \underline{\text{fi}} \\
= & \{ \beta(w, v) \leq \beta(w, w) \equiv v = w \text{ and } \beta(w, w) = 1, \text{ from (5)} \} \\
& \underline{\text{if}} v = w \rightarrow 1 \\
& \square v \neq w \wedge \beta(m[v].s, v) \leq \beta(m[v].s, w) \rightarrow \beta(m[v].s, v) + 1 \\
& \square v \neq w \wedge \beta(m[v].s, v) > \beta(m[v].s, w) \rightarrow \infty \\
& \underline{\text{fi}} \\
= & \{ v \neq w \text{ and } \beta(m[v].s, w) = \infty, \text{ from (14) and (16)} \} \\
& \beta(m[v].s, v) + 1 \quad ,
\end{aligned}$$

and similarly

$$\beta(w, \text{nil})(m := (m; w, s: m[v].s)) = \beta(m[v].s, \text{nil}) + 1 \quad .$$

For $0 \leq i < \beta(u, v)$,

$$\begin{aligned} & (m[\alpha(i, u)].c)(m := (m; w, s: m[v].s)) \\ = & \{(3); \text{Theorem 0 with } p, q := w, m[v].s\} \\ & \underline{\text{if}} \ 0 \leq i < \beta(u, w) \rightarrow m[\alpha(i, u)].c \\ & \square \ \beta(u, w) \leq i < \beta(u, v) \\ & \rightarrow m[\alpha((i - \beta(u, w)) \bmod \beta(m[v].s, w), m[v].s)].c \\ & \underline{\text{fi}} \\ = & \{(16)\} \\ & m[\alpha(i, u)].c \end{aligned}$$

and for $0 \leq i < \beta(m[v].s, \text{nil})$,

$$\begin{aligned} & (m[\alpha(i, w)].c)(m := (m; w, s: m[v].s)) \\ = & \{(3); \text{Theorem 0 with } u, p, q := w, w, m[v].s\} \\ & \underline{\text{if}} \ 0 \leq i < \beta(w, w) \rightarrow m[\alpha(i, w)].c \\ & \square \ \beta(w, w) \leq i < \beta(m[v].s, \text{nil}) \\ & \rightarrow m[\alpha((i - \beta(w, w)) \bmod \beta(m[v].s, w), m[v].s)].c \\ & \underline{\text{fi}} \\ = & \{\beta(m[v].s, w) = \infty, \text{ from (14) and (16)}; \\ & \beta(w, w) = 1, \text{ from (5)}; \\ & \alpha(0, w) = w, \text{ from (4)} \\ & \} \\ & \underline{\text{if}} \ i = 0 \rightarrow m[w].c \\ & \square \ 1 \leq i < \beta(m[v].s, \text{nil}) \rightarrow m[\alpha(i-1, m[v].s)].c \\ & \underline{\text{fi}} \quad . \end{aligned}$$

It follows that

$$\begin{aligned} & wp(w^{\wedge}.s := v^{\wedge}.s; v^{\wedge}.s := w, \text{seq}(u) = S) \\ = & \{\text{Example 3 with } p, q := v, w\} \end{aligned}$$

$$\begin{aligned}
& \text{wp}(w^{\cdot}s := v^{\cdot}s, (17)) \\
= & \{ \text{substitution of the results derived above, using (14)} \} \\
& \beta(m[v].s, v) = \infty \\
& \wedge (\text{SEQ } i: 0 \leq i < \beta(u, v): m[\alpha(i, u)].c) ++ \langle m[w].c \rangle \\
& \quad ++ (\text{SEQ } i: 1 \leq i < \beta(m[v].s, \text{nil}): m[\alpha(i-1, m[v].s)].c) = S \\
= & \{ \text{first conjunct is implied by second on account of Lemma 4 and (13);} \\
& \quad \text{dummy transformation, } i := i+1, \text{ in second sequence} \\
& \} \\
& (\text{SEQ } i: 0 \leq i < \beta(u, v): m[\alpha(i, u)].c) ++ \langle m[w].c \rangle \\
& ++ (\text{SEQ } i: 0 \leq i < \beta(m[v].s, \text{nil}) - 1: m[\alpha(i, m[v].s)].c) = S \quad .
\end{aligned}$$

With the notations (6), (10), (12) the result we have now obtained can be written as follows: under assumption of (13) through (16),

$$\begin{aligned}
& \text{wp}(w^{\cdot}s := v^{\cdot}s; v^{\cdot}s := w, \text{seq}(u) = S) \\
\equiv & \text{seq}_v(u) ++ \langle w^{\cdot}c \rangle ++ \text{seq}(v^{\cdot}s) = S \quad .
\end{aligned}$$

In other words, $w^{\cdot}c$ is inserted into $\text{seq}(u)$ after $v^{\cdot}c$.

References

- [0]H. van Evert, Semantiek van pointers in Pascal (in Dutch). Master's thesis, Eindhoven University of Technology, 1988.
- [1]D. Gries & G. Levin, Assignment and procedure call proof rules. ACM Trans. Program. Lang. Syst. 2 (1980), 564-579.
- [2]C.A.R. Hoare & N. Wirth, An axiomatic definition of the programming language PASCAL. Acta Inf. 2 (1973), 335-355.
- [3]J.W. Hunt & T.G. Szymanski, A fast algorithm for computing longest common subsequences. Comm. ACM 20 (1977), 350-353.
- [4]J.M. Morris, A general axiom of assignment; in: Theoretical foundations of programming methodology (lecture notes international summer school, Marktoberdorf, 1981). NATO Adv. Sci. Inst. Ser. C, Math. Phys. Sci. 91. Reidel, Dordrecht, 1982; pp. 25-34.
- [5]J.M. Morris, Assignment and linked data structures. Ibid., pp. 35-41.
- [6]A.M. de A. Price, Defining dynamic variables and abstract data types in Pascal. ACM

- SIGPLAN Notices 19 (1984), □2, 85-91.
- [7]H.C. Thacher, Jr., On the elimination of pointer variables and dynamic allocation in higher level languages. ACM SIGPLAN Notices 19 (1984), □4, 44-46.
- [8]N. Wirth, On the design of programming languages, in: J.L. Rosenfeld (ed.), Information processing 74. North Holland, Amsterdam, 1975; pp. 386-393.

January, 1988

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