Correspondence

Formal Specification

Dear Sir,
Please allow me not to be convinced that Duce and Fielding's definition[1] of 'empty_...' in the middle of the right-hand column of p. 317 as one of the **ops** guarantees the existence of an element 'empty' as ascertained in the lower part of the column, always different from other elements covered by OBJ; see for instance a primitive finite cyclic group as representative of *NDC_POINTS* and map 'empty to the unity of that group
'_::_' to the multiplier/combiner of that group.
In addition allow a small critique against using '_' for two contradictory purposes:

(1) concatenate name parts in order to avoid name delimiters
(2) announce argument positions within operator names

so: *mk*_ndc_point
↑

does not mean an operation with one (or two?) infix positions even if I might be tempted to write

mk HIMALAYA ndc: 1 *point*

with *HIMALAYA* being (8882). Otherwise I applaud your entertaining explanations and clarifications, enlarging (at least my) knowledge.

Very truly your humble reader
Yours sincerely
K. BROKATE
Rübezahlweg 20,
D-725 Leonberg,
F.R.G. 11.8.87

Reference

1. D. A. Duce and E. V. C. Fielding, Formal specification – a comparison of two techniques. *The Computer Journal* **30** (4), 316–327 (1987).

Termination Indicators in Programming

Dear Sir,
In A. W. Bowker's letter,[1] both programs displayed in the note for a bubble sort are incorrect:

The 6. line states sorting in ascending sequence as goal and requires existence of a[0] and of a[N] by line 3, 1 for j = 0 and line 10, 5, 3, 1 for j = N−1; so N+1 is the count of elements. For 2 elements: N = 1, and there is *no* swapping by line 2, even if a[0] > a[1]; the result is *not* sorted in ascending sequence.

There is one more error in the second program: The 5. line restricts swapping to j < N−1; there is *no* swapping for j = N−1, even if a [N−1] > a[N]; the result is *not* sorted in ascending sequence at the last element. And this in spite of the 12. line, as its reduction t: = s: = j is only possible when j<t before s: = j and this before j: = j+1, so t can only decrease from N−1.

Really it is a tricky business to get a clear "algorithm" transformed into a minimally demanding and totally correct program of atomic steps with iteration:

Always be sure to have proved the correctness with all possible combinations of the input in its domain!

In addition the first program does not restrict the iteration line 5..10 by j < N, so it will run into outside accesses and destroy its environment, if not caught by security measures of the executor.

Yours sincerely
KLAUS BROKATE
Rübezahlweg 20,
D-725 Leonberg,
F.R.G.

Reference

1. A. W. Bowker, Termination indication in programming, *The Computer Journal* **30** (2), 191 (1987).

Mrs Bowker comments

It is quite true that the algorithms will not sort $a[N]$, nor are they intended to do so, as they are a direct modification of Er's algorithm, which is clearly stated to be sorting N elements in array $a[0..N-1]$.

If we are sorting $N+1$ elements in $a[0..N]$ the first line becomes $t := N$; and the algorithms will work correctly. (Similarly, if we are sorting $a[1..N]$ all the zeros must become ones.)

Obviously it would have been better to use the notation $t := hi$;

while $t > lo$ **do begin**
$j := lo$

etc. to sort $a[lo..hi]$. A convention to consider as the norm for future publications perhaps?

There is no danger of outside access in either algorithm. In fact in the first version $j < i$ where $i := t$ (which is $N-1$ (*hi*) at most), I only left the $j \neq i$ to make it more comparable with Er's version, but $j > i$ cannot happen because j is incremented by one on each iteration and $j = i$ will stop it.

Corrigendum: The Reve's Puzzle by Rohl and Gedeon

Sir,

In vol. **29**, no. 2 of *The Computer Journal* you published a paper by T. D. Gedeon and myself entitled 'The Reve's Puzzle'. Unfortunately, in the final draft we left out a crucial sentence. In the second last paragraph we describe a function $F(m, n)$ which returns the number of cheeses to be moved using m stools, given that n are required to be moved altogether. The following sentence should have been added after that which ends '...kth element of the column to its left'.

For n greater than the sum of these elements, we use n minus the kth element of the mth column.

The significance of this omission is not that the function as originally published is wrong – it's not – but that it does not lead to an optimal solution. The procedure we used, which incorporated the refinement above, is optimal, as our comment in the last paragraph of the paper claimed.
Yours faithfully

J. S. ROHL
Department of Computer Science,
The University of Western Australia,
Nedlands, WA 6009, Australia

Announcement

17–21 OCTOBER 1988

Third International Symposium on Knowledge Engineering, Madrid, Spain

In his opening lecture at the First International Symposium on Knowledge Engineering, Professor H. Simon remarked that any experimental science, to develop, needs an array of tools to attain its goals.

This is even more appropriate to the field of artificial intelligence in general and knowledge engineering in particular since, according to Bacon, 'Relinquished hands and knowledge, by themselves, hardly have any strength. Effects are produced by means of instruments and understanding needs them no less than hands.'

The importance of tools, as derived from the above paragraphs, applies both to knowledge engineering hardware and software. This Third International Symposium is devoted mainly to the discussion of these topics. However, other important topics will also be considered, among them the following.

- Cognitive modelling.
- Commonsense reasoning.
- Acquisition and representation of knowledge.
- Interfaces: natural language, computer vision, etc.
- Knowledge engineering and education.
- Philosophical basis and impact of the technology.

Additional information may be obtained from: Mr José R. Chelala, Third International Symposium on Knowledge Engineering, Alvarez de Baena 3-2°, 28006 Madrid, Spain.