Correspondence

Dear Sir,

The Tower of Hanoi as a Trivial Problem—A Reply

Heard makes a few interesting remarks in his correspondence[1] concerning my recent paper.[2] In replying to his criticism, I wish to clarify a few issues.

(1) I use the word 'intractable' in the conclusion section for the following reason: Hayes,[3] Buneman and Levy[4] were not able to prove Property 3, namely the smallest disc always moves in a cyclic order—a property they used critically in their iterative algorithms. If the Tower of Hanoi problem was not 'intractable', why could they not prove such an essential property on which their algorithms relied?

(2) Heard argues: 'the trivial nature of the problem leads to a lack of appreciation of the power of recursion . . .'. The reason he says that the Tower of Hanoi is a trivial problem is because he has seen a solution to it in the literature. Suppose he was asked to write an algorithm to solve the Tower of Hanoi problem without having seen a solution; it is doubtful that he would find the problem trivial. Moreover, I do not believe that the above-quoted statement is correct. At least from my own teaching experience, the students seem to appreciate the power of recursion but find it not so easy to apply. Indeed, the complexity of recursion in problem-solving has long been overlooked in the literature.[5]

(3) Heard criticizes that I analyzed the *moves* of the discs but not the *status* of the discs after each move. Well, in a representation one must be careful to make explicit only the information needed for solving a problem. As the moves of the discs are more important than the status of the discs after each move in solving the Tower of Hanoi problem, I thus make explicit the disc moves in the representation, and also make precise statements about them. The status of the discs is of no concern in solving the problem and in deriving an algorithm for it.

(4) Although Heard has given an equation for computing 'the peg on which disc j resides after move x', he still has an obligation to prove that the equation and the tabulation of 'each disc which peg it resides on after each move' are correct. For instance, he has yet to show that his tabulation depicts all optimal solutions before making a claim that 'all the properties of the Tower of Hanoi problem follow trivially'. It is not obvious that Property 5 follows from his equation. At any rate, his equation cannot be used efficiently in an algorithm for the Tower of Hanoi problem. His equation involves exponentiation and has to be used costly for calculating the status of each disc per cycle of move.

Interestingly, since the publication of the paper,[2] I have discovered the following theorem which makes the Tower of Hanoi problem really trivial:

Theorem. Assuming all the discs are numbered from 1 to n with the smaller disc being assigned the smaller number, the disc to be moved in step x is precisely the position index of the rightmost 1 in the binary numeral of x.

Proof. Omitted.

Coupled with Property 6, the above theorem can be used to solve the Tower of Hanoi problem trivially. However, what puzzles us is that why it takes the mankind a *century* to discover such a simple solution.

Yours faithfully

M. C. ER
Department of Computing Science
University of Wollongong
PO Box 1144
Wollongong
New South Wales 2500
Australia

References

1. R. J. Heard, The Tower of Hanoi as a trivial problem. *The Computer Journal* **27**, 90 (1984).
2. M. C. Er, A representation approach to the Tower of Hanoi problem. *The Computer Journal* **25**, 442–447 (1982).
3. P. J. Hayes, A note on the Towers of Hanoi problem. *The Computer Journal* **20**, 282–285 (1977).
4. P. Buneman and L. Levy, The Towers of Hanoi problem. *Information Processing Letters* **11**, 243–244 (1980).
5. M. C. Er, On the complexity of recursion in problem solving. *International Journal of Man–Machine Studies*, in press.

Dear Sir,

Security Management and Protection—A Personal Approach

[M. V. Wilkes, *The Computer Journal* **27**, 3 (1984)]

I read with interest Mr Wilkes' article 'Security Management and Protection—a Personal Approach' in the February 1984 issue of the Journal.

I wish to point out that the statement that 'auditors issue a certificate confirming the correctness of a set of accounts' is an incorrect understanding of the statutory duties laid down for auditors. The requirement that a set of accounts be certified as 'correct' was withdrawn with the introduction of the Companies Act 1947 (consolidated into the 1948 Act) and in its place auditors are required to report whether the set of accounts shows or does not show 'a true and fair view of the state of the affairs' of the entity being audited. Similar requirements exist in other Acts of Parliament governing the statutory audits of entities other than Limited Companies.

Whilst this point does not detract from Mr Wilkes plea for certifiable software it does underline an unfortunate misunderstanding amongst many people as to the role and duties of auditors.

Yours faithfully

W. LIST
(Chairman of the British
Computer Society Audit
by Computer Specialist Group)
Thomson McLintock & Co
70 Finsbury Pavement
London EC2A 1SX
UK

Dear Sir,

User-Defined Types in a Polymorphic Language

[D. M. Harland, *The Computer Journal* **27**, 47 (1984)]

The author states in Section 4 of his paper that Strachey did not consider giving 'civil rights' to types. This is not quite true. Among my souvenirs I have a draft CPL manual of 1965 vintage in which there is a type type and also a type general (Section 3.2.4 for any other hoarders of such documents). I also recall hearing Strachey lecture on the idea in Oxford at around the same time. I am therefore certain that Strachey *et al.* were actively considering polymorphism.

Yours faithfully

DAVID HOLDSWORTH
University Computing Service
The University of Leeds
Leeds LS2 9JT
UK

Unfortunately, Christopher Strachey is no longer with us and cannot comment on the points made by David Holdsworth. However, one of Christopher Strachey's coworkers on CPL, Professor David Barron now of the University of Southampton, made the following comment.

David Barron replies

Dr Holdsworth is quite correct. I well remember Christopher Strachey expounding his ideas on types, which led to the inclusion of the types 'type' and 'general' in CPL. We were actively considering polymorphism, though at the time we used the term 'polymorphic' in the sense that ADA uses 'generic'.