
CORRESPONDENCE

‘What is SP?’: A Reply

J. GERARD WOLFF

*School of Electronic Engineering and Computer Systems, University of Wales, Dean Street, Bangor,
Gwynedd, LL57 1UT, UK*

My interest in information compression as an explanatory principle in information processing was sparked originally by lectures given by Horace Barlow in Cambridge University describing phenomena in the workings of brains and nervous systems which could be understood in terms of economical coding of information (see, for example, [1] and [2]). Later, an accumulation of evidence led me to the view that many aspects of language learning by children could be understood in these terms [3].

About 1987, I began to realize that several aspects of ‘mainstream’ computing might also be understood as information compression. These insights suggested the possibility of developing a new kind of computing system with more flexibility and ‘intelligence’ than conventional computers. An article describing this thinking was published in this journal.

Subsequent research has attempted to resolve the many questions which naturally arise from the ‘SP’ conjecture that *all kinds of computing and formal reasoning may usefully be understood as information compression by pattern matching, unification and metrics-guided search*. Recent publications in this programme of research include [4], which describes the background and motivation for the research, [5] which describes how the design of software and its execution may be seen as information compression, [6] which describes an application of these ideas in best-match retrieval of information, [7] which is an overview of the research at mid-1994 and [8] which describes a recent model of the proposed new system. The latest thinking is described in [9], [10] and [11].

1. A RESPONSE

The purpose of this short article is to respond to an earlier article in this journal [12] which criticizes the programme of research. That article is very similar to another article by the same two authors published in another journal [13]. Both articles are riddled with inaccuracies and misconceptions.

A detailed response to [12] is not necessary here because a full response to the earlier article has already been published [14] and it answers all the main points in [12]. The previous reply has been written so that it is free standing and may be understood without reference to earlier publications. Readers with an interest in these

matters should find that it provides useful commentary on aspects of the research.

In the rest of this article I will briefly discuss three themes which seem to deserve more comment: questions of research strategy and ‘vision’, computational complexity, and determinism.

2. RESEARCH STRATEGY AND VISION

Like the earlier article, [12] demonstrates a basic misunderstanding of the strategy in this programme of research.

From the beginning, there has been the germ of an idea which has suggested the possibility of substantial benefits in terms both of science and of applications. These potential benefits constitute a ‘vision’ of where the research may lead which provides the motivation for doing the research. They cannot possibly be ‘claims’ for the SP theory because the central idea in that theory is still only a conjecture and has been clearly marked as such from the beginning.

Given the motivation for pursuing these ideas, the strategy has been to adopt the SP conjecture as a working hypothesis and to see how far it can be ‘pushed’. If it turns out to be wrong, relatively little is lost. If it turns out to be right, the potential benefits are very large.

The central idea is embodied in a ‘model’. Each area of potential application needs to be examined to see whether the model can be applied and, if so, how. Apparent contradictions or conflicts in the ideas need to be examined to see whether they provide sufficient reason to abandon the hypothesis or whether they can be resolved in other ways.

The model is progressively refined to accommodate as wide a range of concepts as possible. At all stages, the temptation to add *ad hoc* features to the model must be resisted. In accordance with good practice in science and engineering and, indeed, in accordance with the SP theory itself, the aim at all stages is to develop a model which combines simplicity with explanatory or descriptive ‘power’.

The authors of [12] fail to recognize this process of progressive refinement and the need to preserve parsimony in the developing model. There is an implicit belief that a theory springs into life fully formed with all its details in place. The article assumes that elements of

the model like 'variable' or 'negation' can be given arbitrary formal definitions without regard to the overall framework of ideas.

The uncertainty about concepts like 'variable' and 'negation' has arisen because it has not been clear whether they could be accommodated as emergent properties of the core model or whether they should be adopted as primitives in the theory. I am now reasonably confident that there is no need for a concept of a variable as a primitive component of the model. Until recently it has appeared that a concept of negation would be required as a primitive but recent insights suggest that, like the concept of a variable, negation may be modelled by other constructs.

3. COMPUTATIONAL COMPLEXITY

The authors of [12] seem not to have understood that there are many interesting problems in computing, especially in the area of artificial intelligence, where the abstract space of possible solutions is too large to be searched exhaustively. They seem not to have understood, either, that there are standard ways of dealing with this type of problem which are described in every elementary text book of artificial intelligence.

The basic idea is to search the solution space in stages, choosing the most promising path or paths at each stage and thus ignoring large parts of the space. With this approach, reasonably good solutions can often be found without undue computational effort but ideal solutions cannot normally be guaranteed.

There is a trade off between computational complexity and the effectiveness of the search. However large the search space may be, there is always the possibility of searching it in such a way that the computational complexity of the process is acceptable. This normally means some sacrifice in the effectiveness of the search but, for many practical problems, acceptably good solutions can often still be found.

These ideas are found in different forms in a range of techniques including hill climbing (descent), beam search, genetic algorithms, simulated annealing and others. These kinds of techniques may be described collectively as metrics-guided search.

3.1. Towards a general-purpose search method

The SP programme aims to develop a theory of computing which embraces AI problems as well as the more traditional kinds of problem. That being so, the theory must necessarily embrace principles of metrics-guided search.

At present, this kind of search appears in many different programs and systems. If it is true that *all* kinds of computing and formal reasoning may be understood in terms of pattern matching, unification and metrics-guided search then there is the interesting possibility of developing *one* general-purpose search method which may serve many different purposes.

This idea is similar to the way in which search mechanisms in a database management system or expert system shell may be used for diverse applications thus saving the need to reprogram those search mechanisms for each application. The SP programme seeks to push this idea further by generalizing existing search methods to integrate learning, reasoning, information retrieval and other aspects of computing within one simple conceptual framework.

4. DETERMINISM

The remarks about determinism in WSP appear to reflect a naïf view of the nature of computing and formal systems. The authors appear to be looking for clockwork certainty in formal systems long after Gödel provided a negative answer to the question posed by Hilbert about whether such certainty was possible. Again: '*I have recently been able to take a further step along the path laid out by Gödel and Turing. By translating a particular computer program into an algebraic equation of a type that was familiar even to the ancient Greeks, I have shown that there is randomness in the branch of pure mathematics known as number theory. My work indicates that—to borrow Einstein's metaphor—God sometimes plays dice with whole numbers!*' ([15] p. 80).

A Turing machine may be deterministic or non-deterministic depending on its transition function and the information on its tape. In a similar way, it appears that the proposed SP system may be deterministic or non-deterministic depending on its search method and the information with which it is supplied. But none of these systems can escape from Gödel's result: all computational systems are necessarily fuzzy at the edges.

5. CONCLUSION

As already indicated, detailed answers to the main points in [12] are given in [14] which may be read as a free-standing commentary on aspects of the research. However, I hope that readers will judge for themselves the merits of these ideas by consulting other sources which have been cited. The best introduction is probably [7] and [8]. Background thinking is described in [4].

REFERENCES

- [1] Barlow, H. B. (1969) Trigger features, adaptation and economy of impulses. In Leibovic, K. N. (ed.), *Information Processing in the Nervous System*, Springer, New York, pp. 209–230.
- [2] Barlow, H. B. and Földiák, P. (1989) Adaptation and decorrelation in the cortex. In Durbin, R. M., Miall, C. and Mitchison, G. J. (eds), *The Computing Neuron*, Addison-Wesley, Wokingham, UK, pp. 54–72.
- [3] Wolff, J. G. (1988) Learning syntax and meanings through optimization and distributional analysis. In Levy, Y., Schlesinger, I. M. and Braine, M. D. S. (eds), *Categories and Processes in Language Acquisition*, Lawrence Erlbaum, Hillsdale, NJ (reproduced in [16], Chapter 2), pp. 179–215.

- [4] Wolff, J. G. (1993) Computing, cognition and information compression. *AI Communications*, 6(2), 107–127.
- [5] Wolff, J. G. (1994a) Towards a new concept of software. *Software Engineering Journal*, 9(1), 27–38.
- [6] Wolff, J. G. (1994b) A scalable technique for best-match retrieval of sequential information using metrics-guided search. *Journal of Information Science* 20(1), 16–28.
- [7] Wolff, J. G. (1995a) Computing as compression: an overview of the SP theory and system. *New Generation Computing*, 13, 187–214.
- [8] Wolff, J. G. (1995b) Computing as compression: SP20. *New Generation Computing*, 13, 215–241.
- [9] Wolff, J. G. (1995c) Learning and reasoning as compression by multiple alignment and unification. Paper presented at Applied Decision Technologies 1995, Brunel, April 1995.
- [10] Wolff, J. G. (submitted, a) Computing as compression by multiple alignment, unification and search (1).
- [11] Wolff, J. G. (submitted, b) Computing as compression by multiple alignment, unification and search (2).
- [12] Stephen, G. A. and Mather, P. (1994) What is SP? *Computer Journal*, 37(9), 745–752.
- [13] Stephen, G. A. and Mather, P. (1993) Sweeping away the problems that dog the industry? *AI Communications*, 6(3/4), 213–218.
- [14] Wolff, J. G. (1994c) Computing and information compression: a reply. *AI Communications*, 7(3/4), 203–219.
- [15] Chaitin, G. J. (1988) Randomness in arithmetic. *Scientific American*, 259(1), 80–85.
- [16] Wolff, J. G. (1991) *Towards a theory of cognition and computing*. Ellis Horwood, Chichester, UK.

Correspondence ‘What is SP?’: A Reply

GRAHAM A. STEPHEN

INTRODUCTION

More fundamental than mathematics—this was how Wolff once described SP. This gives a clue to its indented all-encompassing nature. But what exactly is it? That was the question I set out to answer in the paper [1] Wolff responds to here.

Posited as a ‘universal’ theory of computing, SP is not readily unravelled. It is clad in a mantle of obfuscation, fraught with flawed argument and muddled thinking; and imprecision and inconsistency permeate the hyperbolic cant describing it.

SP has nevertheless been scrutinised (e.g. [2]). Recently, a PhD thesis [3] from Wolff’s own department also investigated the theory. This concluded that ‘*SP is not a feasible model of computation, nor can it be made to be so without violating the fundamental premises underlying it*’. And it showed that ‘*the claims made for SP in the SP literature are wholly unsubstantiated, and that the ‘evidence’ of SP’s suitability for various application areas derives from pathological, inconsistent, and incorrect examples*’.

Wolff admits providing any cogent response to the evidence by citing an earlier paper [4]. But far from answering ‘*all the main points*’, that rambling discourse only adds further confusion to the subject. Here, Wolff discusses his research strategy, complexity, and determinism. Let us take a look at each of these in turn.

RESEARCH STRATEGY

Wolff tries to justify his research strategy. He also disowns the claims made for SP, preferring the term ‘potential benefits’. Contrast this with the numerous actual claims scattered throughout Wolff’s papers. To give but one example: ‘*The SP prototype can demonstrate a number of capabilities... The current list... includes: induction of grammars from examples ...; induction of class hierarchies ... logical deduction ... probabilistic inference; information retrieval...; ‘fuzzy’ pattern recognition; means-end analysis.*’ [5].

To return to the research strategy for SP, its development has been desultory. The emphasis has been on empirical programming, to the detriment of proper theoretical analysis. Programming successive implementations to test each latests ‘insight’ has wasted much time and effort. Often, a little careful forethought, and perhaps a basic grasp of some mathematical fundamentals, would have saved much needless effort.

This is typified by the version known as SP8, which implements a more than exhaustive search of all the subsequences of the input (these are exponential in number). Only *after* SP8 had been designed, written, and tested was it found that its inherently intractable search method would ‘*not scale up easily*’. And for later implementations, this superficial understanding of complexity issues seems to have led to the naive objective of performing exhaustive subsequence searches in ‘*linear or better*’ time.

Throughout the development of SP its conjectures have been adhered to dogmatically. Contradictory evidence has been persistently dismissed summarily or ignored altogether. Thorny aspects of the theory have not been properly addressed; most have been shelved indefinitely. Witness the nonsense of the confusion over ‘*emergent properties of the core model*’ and ‘*primitives in theory*’ when Wolff talks here about variables and negation.

COMPUTATIONAL COMPLEXITY AND DETERMINISM

That it is impractical to search exhaustively the solution spaces of many problems is well known. Wolff recognizes that in such cases only approximate solutions can feasibly be attempted. But he seems unable to accept that the correct operation of an SP system relies on finding ideal solutions by (intractable) exhaustive searching.

SP’s capabilities are said to arise as side-effects of a compression of the input data. The simplistic method adopted involves seeking and merging instances of repeated patterns. These patterns are subsequences of the input, and to guarantee that any particular pair of equivalent patterns are merged requires an exhaustive subsequence search.

The ‘SP search problem’ is thus intractable. Practical implementations must therefore adopt some approximation strategy—Wolff’s elusive general-purpose metrics-guided search. But then pattern merges

necessary for the correct operation of any given application can no longer be guaranteed. When implemented in SP, even simple problems soluble in linear time, say, would become either intractable or only soluble approximately.

As an illustration, consider two 'capabilities' of SP: text retrieval and procedural programming. First, when using the search facility of a word processor, for instance, we would want to find all the occurrences of a given word in a document, quickly. A practical SP system might find them. But then again, it might not. Wolff himself admits that the latest version, SP21, can miss '*matches that people can see*' [6]. Second, what use is a system that can only approximately run a program? Users of real programming languages can be confident of the operation of (correct) programs. Potential users of SP would have no such luxury.

Wolff dodges the issue of determinism; it is unclear how Gödel's incompleteness theorem is relevant. This says that any sufficiently powerful formal system is incomplete, inasmuch as there exists at least one true statement that cannot be derived using the axioms and inference rules of the system. The uncertainty in SP, however, arises from the intractability of the SP problem—will any given pair of matching patterns in the input be merged in a reasonable time (or at all when searched non-exhaustively)? Such patterns could, in principle, be recognized and merged by deterministic means given enough time. NP-completeness therefore bears more directly on the practicability of SP than does the incompleteness of formal systems.

CONCLUSION

As a '*general theory of computing*' [7] SP has little merit. Wolff burdens all potential applications with the

intractability of the 'SP search problem'. And—the impracticability aside—the 'capabilities' of SP are unconvincing, each relying on *ad hoc*, human interpretation of a compressed input text.

SP does not fulfil its grandiose promises: we still await the realization of Wolff's [8] vision of '*a new computing language... driven by a new kind of computer*' exhibiting '*a significant measure of... human-like flexibility*' and bringing about a '*global simplification and streamlining*' of computing.

REFERENCES

- [1] Stephen, G. A. and Mather, P. (1994) What is SP? *The Computer Journal*, 37(9), 745–752.
- [2] Stephen, G. A. (1993) *An Examination of a Conjectured Compression-based Model of Computation*. Technical Report TR-93-gas-01, School of Electronic Engineering and Computer Systems, University of Wales, Bangor, Gwynedd, UK (available from URL <ftp://venus.sees.bangor.ac.uk/pub/graham/>).
- [3] Mather, P. (1994) *An Evaluation of an Extant Proposed New Theory of Computing Based on Information-Theoretic Principles and Data Compression*. PhD Thesis, School of Electronic Engineering and Computer Systems, University of Wales, Bangor, Gwynedd, UK.
- [4] Wolff, J. G. (1994a) Computing and information compression: a reply. *AI Communications*, 7(3/4), 203–219.
- [5] Wolff, J. G. (1991) The SP theory and the 'symbolic'/'sub-symbolic' distinction. *AISB Quarterly*, 78, 40–42.
- [6] Wolff, J. G. (1994b) A scaleable technique for best-match retrieval of sequential information using metrics-guided search. *Journal of Information Science*, 20(1), 16–28.
- [7] Wolff, J. G. (1993) Computing, cognition and information compression. *AI Communications*, 6(2), 107–127.
- [8] Wolff, J. G. (1991) A new theory for old problems. *The Computer Bulletin*, series IV, 1(7), 8–9.