# Mathematics as information compression via the matching and unification of patterns

J Gerard Wolff*

September 21, 2019

## Abstract

This paper describes a novel perspective on the foundations of mathematics: how mathematics may be seen to be largely about 'information compression (IC) via the matching and unification of patterns' (ICMUP). That is itself a novel approach to IC, couched in terms of non-mathematical primitives, as is necessary in any investigation of the foundations of mathematics. This new perspective on the foundations of mathematics reflects the idea that, as an aid to human thinking, mathematics is likely to be consonant with much evidence for the importance of IC in human learning, perception, and cognition. This perspective on the foundations of mathematics has grown out of a long-term programme of research developing the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*, a system in which a generalised version of ICMUP—the powerful concept of *SP-multiple-alignment*—plays a central role. The paper shows with an example how mathematics, without any special provision, may achieve compression of information. Then it describes examples showing how variants of ICMUP may be seen in widely-used structures and operations in mathematics. Examples are also given to show how several aspects of the mathematics-related disciplines of logic and computing may be understood as ICMUP. Also discussed is the intimate relation between IC and concepts of probability, with arguments that there are advantages in approaching AI, cognitive science, and concepts of probability via ICMUP. Also discussed is how the close relation between IC and concepts of probability relates to the established view that some parts of mathematics are intrinsically probabilistic, and how that

---

*Dr Gerry Wolff, BA (Cantab) PhD (Wales) CEng MBCS MIEEE, ORCID ID 0000-0002-4624-8904. CognitionResearch.org, Menai Bridge, UK; Visiting Researcher at Coventry University; jgw@cognitionresearch.org; +44 (0) 1248 712962; +44 (0) 7746 290775; *Skype*: gerry.wolff; *Web*: www.cognitionresearch.org.

1

latter view may be reconciled with the all-or-nothing, 'exact', forms of calculation or inference that are familiar in mathematics and logic. There are many potential benefits and applications of the mathematics-as-IC perspective.

*Keywords:* Foundations of mathematics; information compression; SP Theory of Intelligence.

# 1  Introduction

The fundamental nature of mathematics has been a considerable puzzle to mathematicians and others for many years. For example, Roger Penrose writes:

> "It is remarkable that *all* the SUPERB theories of Nature have proved to be extraordinarily fertile as sources of mathematical ideas. There is **a deep and beautiful mystery** in this fact: that these superbly accurate theories are also extraordinarily fruitful simply as *mathematics*."
> ([45, pp. 225–226], bold face added).

In a similar vein, John Barrow writes:

> "**For some mysterious reason** mathematics has proved itself a reliable guide to the world in which we live and of which we are a part. Mathematics works: as a result we have been tempted to equate understanding of the world with its mathematical encapsulization. ... **Why is the world found to be so unerringly mathematical?**"[1]  ([7, Preface, p. vii], bold face added).

And Eugene Wigner [59] writes about "The unreasonable effectiveness of mathematics in the natural sciences":

> "The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which **we neither understand** nor deserve. We should be grateful for it and hope that it will remain valid in future research and that it will extend, for better or for worse, to our pleasure, even though perhaps also to

---

[1]It is clear that, in this quote, the expression "the world" is intended to mean "everything in the observable universe", in accordance with normal usage. That expression is intended to have the same meaning elsewhere in this paper.

2

**our bafflement**, to wide branches of learning."[2] (*ibid*, p. 14, bold face added).

In keeping with those remarks, Øystein Linnebo writes that: "Mathematics poses a daunting philosophical challenge, which has been with us ever since the beginning of Western philosophy." [33, p. 4]. He goes on to say: 1) that mathematics is *a priori* because it seems to be practiced by means of reflection and proof alone, without any reliance on sense experience or experimentation; 2) that mathematics seems to deliver knowledge of truths that are *necessary* in the sense that things could not have been otherwise; and 3) that mathematical knowledge is *abstract*, being concerned with objects such as numbers, sets, and functions, that are not located in space or time, and that don't participate in causal relationships. "In short, by being so different from the ordinary empirical sciences, mathematics is philosophically puzzling; but simultaneously, it is rock solid." [33, pp. 4–5].

This paper attempts to provide some answers. It describes how much of mathematics, perhaps all of it, may be seen as structures and processes for compressing information via a search for patterns that match each other and by the merging or unifying patterns that are the same.[3] This perspective appears to be novel, not apparently described anywhere in writings about the fundamentals of mathematics (Section 2).

The ideas and arguments presented in this paper have grown out a long-term programme of research developing the *SP System*, meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*, both of them outlined in Section 3, and both of them founded on evidence that information compression (IC) is a unifying principle in much of human learning, perception, and cognition (HLPC).

What appear to be the most compelling kinds of evidence for the importance of IC in HLPC are described in [71, Sections 4 to 21]. Examples include: the mismatch between the relatively large volumes of information reaching the retina of the eye and the relatively small capacity of the optic nerve to transmit that information, with evidence for compression of information in the eye; the way in

---

[2]In this connection: "... against Wigner's 'unreasonable effectiveness' statement (based on success in the physical sciences) one must ask why maths is often so unreasonably ineffective in the human and social sciences of behaviour, psychology, economics, and the study of life and consciousness. These complex sciences are dominated by non-linear behaviour and only started to be explored effectively by many people (rather than only huge well-funded research groups) with the advent of small personal computers (since the late 1980s) and the availability of fast supercomputers. Some complex sciences contain unpredictabilities in principle (not just in practice): predicting the economy changes the economy whereas predicting the weather doesn't change the weather." John Barrow, personal communication, 2017-04-06, with permission.

[3]This paper draws on and considerably expands and refines some of the thinking in [61, Chapter 10].

which we merge successive views of a scene to make one; how recognition may be seen as a merging of sensory information with already-stored information; how people with two functioning eyes merge the two simultaneous views of a scene from the two eyes into a single view; how natural language provides an abundance of examples of the 'chunking-with-codes' technique for compression of information; and more.

In view of that evidence, and since mathematics has been developed almost exclusively by human brains and as an aid to human thinking, it should not be surprising that mathematics may be founded on compression of information.

The main sections which follow are: a summary of some writings about the foundations of mathematics, with a description of the novelty of the idea that mathematics may be understood in terms of IC (Section 2); an outline description of the SP System and its foundations (Section 3); a summary of some related research (Section 4); a description of seven techniques for compression of information which are central in the arguments in the sections (Section 5); the main subject of this paper: how mathematics may be interpreted in terms of IC (Section 6); how similar principles may be seen in the mathematics-related disciplines of logic and computing (Section 7); some remarks about the intimate relation between IC and concepts of probability (Section 8); and Section 9 outlines some potential benefits and applications of the ideas which have been described; Appendix A describes two apparent contradictions of the idea that IC is fundamental in mathematics and related disciplines, and how those apparent contradictions may be resolved.

## 2 Writings about the foundations of mathematics

This section first describes the more prominent 'isms' in the philosophy of mathematics, and then describes the novelty of the idea that mathematics may be understood in terms of IC.

### 2.1 Isms in the philosophy of mathematics

The variety of 'isms' in the philosophy of mathematics testifies to the difficulty of arriving at a satisfactory account of the fundamental nature of mathematics. The more prominent of those isms are summarised alphabetically here:

- *Formalism.* Linnebo writes: "*Formalism* is the view that mathematics has no need for semantic notions, or at least none that cannot be reduced to syntactic ones." [33, p. 39]. He goes on to describe two versions of formalism and another variant called *deductivism*:

- *Game Formalism.* "One version of formalism latches on to the comparison of a formal proof with a game played with syntactic expressions. According to *game formalism*, this is all there is to mathematics. That is, mathematics revolves around formal systems, which are syntactical games played with meaningless expressions." ([33, p. 39], emphasis in the original).

- *Term Formalism.* "As we [define] it, formalism seeks either to banish all semantic notions from mathematics or else to reduce any such notions to purely syntactic ones. While game formalism pursues the former alternative, *term formalism* pursues the latter. Mathematical singular terms are now allowed to denote themselves." ([33, p. 44], emphasis in the original). The gist of what Lennebo says to explain the idea is that something like '6' or '22' is not simply a pattern on a piece of paper, it is a pattern with an associated meaning.

- *Deductivism.* "*Deductivism* (sometimes also known as *if-then-ism*) is the view that pure mathematics is the investigation of deductive consequences of arbitrarily chosen sets of axioms in some formal and uninterpreted language." ([33, p. 48], emphasis in the original).

- *Hilbert's ideas.* Linnebo describes David Hilbert's ideas about the nature of mathematics like this:

  "The most sophisticated development of formalist ideas is that of Hilbert's program. Hilbert proposes ... a brilliant strategy of divide and conquer. The way forward, he thinks, is to distinguish mathematics into two parts. *Finitary mathematics* is a contentful theory of finite and quasi-concrete syntactic types. Hilbert is particularly fond of numerals that take the form of strings of strokes; for example, '|||' is the third numeral. Such numerals are sequences of what we may call Hilbert strokes. Hilbert thinks that finitary mathematics and its foundational axioms can be accounted for using ideas from Kant and term formalism. *Infinitary mathematics*, on the other hand, is strong enough to describe all of the infinite structures that modern mathematics studies. This part of mathematics can be regarded as a purely formal theory, Hilbert thinks, and when so regarded, can be accounted for by drawing on ideas from game formalism." ([33, p. 56], emphasis in the original).

Linnebo goes on to discuss problems for Hilbert's program associated with Cantor's ideas about infinities in mathematics and Gödel's incompleteness theorems.

- *Holism.* About holism, Michael Resnik writes: "The observational evidence for a scientific theory bears upon the theoretical apparatus as a whole rather than upon individual component hypotheses." [47, Location 550], and "... we can construct a so-called indispensability argument for mathematical realism along these lines: mathematics is an indispensable component of natural science; so, by holism, whatever evidence we have for science is just as much evidence for the mathematical objects and mathematical principles it presupposes as it is for the rest of its theoretical apparatus; whence, by naturalism, this mathematics is true, and the existence of mathematical objects is as well-grounded grounded as that of the other entities posited by science." (*ibid.*).

- *Intuitionism.* Leon Horsten writes: "Intuitionism originates in the work of the mathematician L. E. J. Brouwer [57], and it is inspired by Kantian views of what objects are [40, Chapter 1]. According to intuitionism, mathematics is essentially an activity of construction. The natural numbers are mental constructions, the real numbers are mental constructions, proofs and theorems are mental constructions, mathematical meaning is a mental construction ... Mathematical constructions are produced by the ideal mathematician, i.e., abstraction is made from contingent, physical limitations of the real life mathematician." [25, Section 2.2].

- *Logicism.* Linnebo writes: "Frege's philosophy of mathematics combines two tenets. On the one hand, he was a platonist, who believed that abstract mathematical objects exist independently of us. On the other hand, he was a logicist, who took arithmetic to be reducible to logic." [33, p. 21]. And Horsten writes:

  "The idea that mathematics is logic in disguise goes back to Leibniz. But an earnest attempt to carry out the logicist program in detail could be made only when in the nineteenth century the basic principles of central mathematical theories were articulated (by Dedekind and Peano) and the principles of logic were uncovered (by Frege). ... In a famous letter to Frege, Russell showed that Frege's Basic Law V entails a contradiction [49]. This argument has come to be known as Russell's paradox ...." [25, Section 2.1].

  An account of Russell's paradox is in [28].

- *Methodological naturalism.* Alexander Paseau writes: "In philosophy of mathematics of the past few decades methodological naturalism has received the lion's share of the attention, so we concentrate on this. ... Methodological

6

naturalism has three principal and related senses in the philosophy of mathematics. The first is that the only authoritative standards in the philosophy of mathematics are those of natural science (physics, biology, etc.). The second is that the only authoritative standards in the philosophy of mathematics are those of mathematics itself. The third, an amalgam of the first two, is that the authoritative standards are those of natural science and mathematics. We refer to these three naturalisms as scientific, mathematical, and mathematical-cum-scientific. Note that throughout this entry 'science' and cognate terms encompass only the natural sciences." [42, Section 1].

- *Nominalism.* Linnebo writes: "In contemporary philosophy of mathematics, 'nominalism' typically refers to the view that there are no abstract objects." [33, p. 101], and:

  "... we need to do to every scientific theory what we did to finite number ascriptions, namely to 'nominalize' the theory by reformulating it in a way that avoids all commitment to abstract objects." [33, p. 105].

- *Platonism.* In the 'Platonism' view, mathematical entities "are not merely formal or quantitative structures imposed by the human mind on natural phenomena, nor are they only mechanically present in phenomena as a brute fact of their concrete being. Rather, they are numinous and transcendent entities, existing independently of both the phenomena they order and the human mind that perceives them." [23, pp. 95–96]. Such ideas are "invisible, apprehensible by intelligence only, and yet can be discovered to be the formative causes and regulators of all empirical visible objects and processes." ([23, pp. 95]).

- *Predicativism.* Horsten writes: "The origin of predicativism lies in the work of Russell. On a cue of Poincaré, he arrived at the following diagnosis of the Russell paradox. The argument of the Russell paradox defines the collection **C** of all mathematical entities that satisfy $\neg x \in x$. The argument then proceeds by asking whether **C** itself meets this condition, and derives a contradiction. ..." [25, Section 2.4].

- *Realism.* Resnik writes: "My realism consists in three theses: (1) that mathematical objects exist independently of us and our constructions, (2) that much of contemporary mathematics is true, and (3) that mathematical truths obtain independently of our beliefs, theories, and proofs. I have used the qualifier 'much' in (2), because I do not think mathematical realists need be

7

committed to every assertion of contemporary mathematics." [47, Location 84].

- *Structuralism.* Linnebo writes: "Structuralism is a philosophical view that emphasizes mathematics' concern with abstract structures, as opposed to particular systems of objects and relations that realize these structures. Consider three children linearly ordered by age and three rocks linearly ordered by mass. These two systems of objects and relations realize the same abstract structure, namely that of three objects in a linear order. All that matters for mathematical purposes, according to structuralism, is the abstract structure of some system of objects and relations, not the particular natures of these objects and relations." [33, p. 154].

There is more about structuralism in Section 4.4.

## 2.2 The novelty of the idea that mathematics may be understood in terms of IC

With regard to the idea that mathematics may be understood in terms of IC, three recent books about the philosophy of mathematics [33, 34, 41], an article about the "Philosophy of Mathematics" in the *Stanford Encyclopedia of Philosophy* [25], two near-recent books in the same area [52, 47], and one recent book on mathematics-related areas [11], make no mention of anything resembling IC. More generally, the idea that IC might be part of the foundations of mathematics appears to have no place in any of the isms in the philosophy of mathematics (Section 2.1), or any other writings about the nature of mathematics.

Devlin's academic book, *Logic and Information* [19], aims to develop a mathematical theory of information, a goal which is related to but distinct from the central idea in this paper, that mathematics may be seen to be largely about IC.

A book for non-specialists by Devlin, called *Mathematics: The Science of Patterns* [18], discusses things like: "patterns of symmetry [such as] the symmetry of a snowflake or a flower" (p. 145) (where 'symmetry' implies redundancy, which is an important part of IC); and "the patterns involved in packing objects in an efficient manner" (p. 152) (where 'efficient' may be seen to relate to IC). But these kinds of pattern are quite different from the concept of an 'SP-pattern' in the SP System (Section 3.2), and IC in the foundations of mathematics is not made explicit or discussed.

Resnik's academic book on *Mathematics As a Science of Patterns* [47] is discussed in Section 4.4, below.

Amongst isms in the philosophy of mathematics (Section 2.1), the one which is perhaps most closely related to the thesis of this paper is *intuitionism*, meaning

8

that mathematics is a creation of the human mind. Clearly, the invention and development of mathematical concepts has been done mainly by human brains, and they are designed to assist human thinking. But there appears to be no recognition in intuitionism of IC as a unifying principle in HLPC or mathematics, of unsupervised learning, or of the representation of knowledge with structures like SP-multiple-alignment.

Concepts in the SP System also relate to structuralism, as discussed in Section 4.4.4.

# 3  Outline of the SP Theory of Intelligence and the SP Computer Model

As noted in the Introduction, much of the thinking in this paper derives from the SP System, meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model*. This section describes the SP System in outline, with sufficient detail to allow the rest of the paper to be understood.

In most papers in the SP programme of research, including this one, it has proved necessary to provide a section like this one, or an appendix, which provides an outline of the SP System. This is to ensure that each paper is free standing and can be read without the need to look elsewhere for information about the SP System.

The most comprehensive account of the SP System is in the book *Unifying Computing and Cognition* [61], which includes a detailed description of the SP Computer Model with many examples of what the Model can do. A shorter but fairly full description of the SP System and its strengths and potential is in [63]. Details of these and other publications, including several papers about potential applications of the SP System, may be found, with download links, on bit.ly/2Gxici2.

Source code and Windows executable code for the SP Computer Model may be downloaded via a link under the heading 'SOURCE CODE' on the same page.

## 3.1  Foundations

The overarching goal in developing the SP System is, in accordance with Ockham's razor, the simplification and integration of observations and concepts across artificial intelligence, mainstream computing, mathematics, and HLPC, with IC as a unifying theme.

Since people often ask what the name "SP" stands for, it is short for *Simplicity* and *Power*. This is 1) because "simplification and integration of observations and concepts" means the same as promoting Simplicity in one's theory whilst retaining

9

as much as possible of its descriptive and explanatory Power; and 2) because within the SP Theory, compression of a body of information, **I**, means maximising the Simplicity of **I** by reducing, as much as possible, repetition of information or *redundancy* in **I**, whilst retaining as much as possible of its non-redundant descriptive or explanatory Power.

Despite the ambition of attempting simplification and integration across AI, computing, mathematics, and HLPC, much has been achieved: the SP System combines Simplicity—in being largely composed of the relatively simple mechanisms for creating and processing SP-multiple-alignments (Section 3.3); and it exhibits descriptive and explanatory Power—in modelling diverse aspects of intelligence, in accommodating diverse kinds of knowledge, and in their seamless integration in any combination, as outlined in Section 3.7.

The idea that IC might be significant in the workings of brains and nervous systems was pioneered by Fred Attneave [1], Horace Barlow [5, 6], and others, and it has been developed in many other studies, many of which are outlined and referenced in [71, Section 3].

## 3.2   The main features of the SP System

As shown schematically in Figure 1, the SP System is conceived as a brain-like system that receives *New* information via its senses and stores some or all of it, in compressed form, as *Old* information.
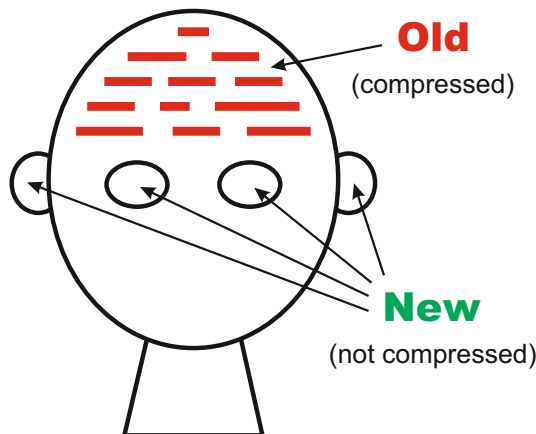


Figure 1: Schematic representation of the SP System from an 'input' perspective. Adapted from Figure 1 in [63], with permission.

In the SP System, all kinds of knowledge are stored in *SP-patterns*, where an SP-pattern is an array of atomic *SP-symbols* in one or two dimensions, where

an 'SP-symbol' is simply a distinctive mark that can be matched in a yes/no manner with any other SP-symbol. At present, the SP Computer Model works only with one-dimensional SP-patterns but it is envisaged that it will be generalised to work with two-dimensional SP-patterns, in addition to 1D SP-patterns. Each SP-pattern and SP-symbol has an associated value for the frequency with which it has occurred in a given body of data.

Although SP-patterns are not very expressive in themselves, they come alive within the framework of *SP-multiple-alignments* (Section 3.3), yielding most of the capabilities summarised in Section 3.7.

## 3.3   SP-multiple-alignment

Two important ideas in the SP System are that:

- IC may be achieved via a search for patterns that match each other and the merging or 'unification' of two or more patterns that are the same. There is more detail in Section 5.

  The expression "IC via the matching and unification of patterns" may be shortened to 'ICMUP'. The expression 'mathematics as ICMUP' may be abbreviated as 'MICMUP'.

- More specifically, IC may be achieved via the concept of *SP-multiple-alignment* which may be seen as a generalisation of six variants of ICMUP, as described in Section 5.7.

The concept of SP-multiple-alignment has been borrowed and adapted from the concept of 'multiple sequence alignment' in bioinformatics. An example of a multiple sequence alignment from bioinformatics is shown in Figure 2. Here, five DNA sequences have been arranged in rows and, by judicious "stretching" of sequences in a computer, matching symbols have been brought into line. A 'good' multiple sequence alignment is one with a relatively large number of matching symbols.

With any multiple sequence alignment that is realistically large, the number of possible alignments is astronomically large. For that reason, it is normally necessary to use heuristic search (hill-climbing or descent) to find alignments that are good, with backtracking to avoid getting stuck on local peaks (hill-climbing), or troughs (descent). With heuristic search one cannot normally prove that the best possible result has been found, but it is normally possible to achieve results that are acceptably good.

```
G G A       G       C A G G G A G G A       T G       G     G G A
| | |       |       | | | | | | | | |       | |       |     | | |
G G | G   G C C C A G G G A G G A       | G G C G     G G A
| | |     | | | | | | | | | | | | |       | |       |     | | |
A | G A C T G C C C A G G G | G G | G C T G G A | G A
| | |       | | | | | | | | | | |   |   |       |     | | |
G G A A     | A G G G A G G A     | A G       G     G G A
| | |       | | | | | | | | |       |   |       |     | | |
G G C A     C A G G G A G G       C   G       G     G G A
```

Figure 2: A 'good' multiple sequence alignment amongst five DNA sequences. Reproduced with permission from Figure 3.1 in [61], with permission.

The key difference between the concept of multiple sequence alignment in bioinformatics and the concept of SP-multiple-alignment is that, in the latter case, a 'good' SP-multiple-alignment is one that allows one New SP-pattern (sometimes more than one) to be encoded economically in terms of one or more Old SP-patterns.

As with the creation of multiple sequence alignments that are good, it is normally necessary to use heuristic techniques, with backtracking where necessary, to find SP-multiple-alignments that are good. As before, with such techniques, it is normally possible to find one or more SP-multiple-alignments that are 'reasonably good' but it is not normally possible to guarantee that the best possible SP-multiple-alignment(s) have been found.

An example of an SP-multiple-alignment is shown in Figure 3. Here, the New SP-pattern is the sentence 't w o k i t t e n s p l a y' shown in row 0. Each of rows 1 to 8 shows one Old SP-pattern representing a grammatical structure, which in each of rows 1, 3 and 5 is a word. The overall effect of the SP-multiple-alignment is to analyse or parse the sentence into its constituent parts, each one marked with its grammatical category.

```
0                        t w o           k i t t e n      s            p l a y           0
                         | | |           | | | | | |      |            | | | |
1                        | | |       Nr 5 k i t t e n #Nr |            | | | |           1
                         | | |           |                |           | | | |
2                        | | |     N Np Nr                 #Nr s #N    | | | |           2
                         | | |       | |                      |       | | | |
3              D Dp 4 t w o #D | |                            |        | | | |           3
               |             | | |                           |        | | | |
4         NP D               #D N |                          #N #NP   | | | |           4
          |                      |                           |        | | | |
5         |                      |                           |     Vr 1 p l a y #Vr     5
          |                      |                           |        |           |
6         |                      |                           | V Vp Vr          #Vr #V   6
          |                      |                           | | |                |
7 S Num       ; NP               |                           #NP V |              #V #S 7
    |         |                  |                                 |
8   Num PL ;                     Np                                Vp                     8
```
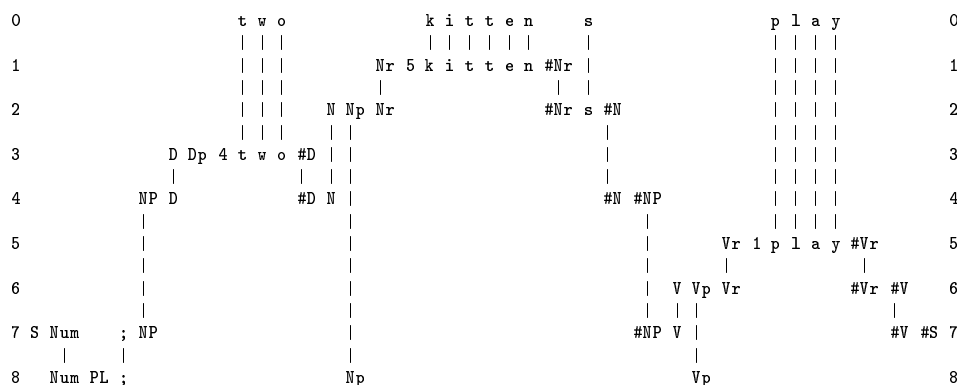
Figure 3: The best SP-multiple-alignment created by the SP Computer Model with a store of Old SP-patterns like those in rows 1 to 8 (representing grammatical structures, including words) and a New SP-pattern, 'two kittens play', shown in row 0 (representing a sentence to be parsed). Adapted from Figures 1 in [62], with permission.

Contrary to what this example may suggest, the concept of SP-multiple-alignment within the SP System can do much more than the parsing of sentences. It is largely responsible for the versatility of the SP System, summarised in Section 3.7. It is not restricted to hierarchical structures as in the parsing example. It can, for example, accommodate discrimination networks and trees, if-then rules, entity-relationship structures, and more. With SP-symbols representing relatively large things such as letters or words, the SP System would have a 'symbolic' flavour, but with SP-symbols representing relatively small things like pixels in an image, the SP System would have a 'non symbolic' flavour.

## 3.4 The calculation of probabilities associated with SP-multiple-alignments

Because of the intimate relation between IC and concepts of probability (Section 8), it is a relatively straightforward matter for the SP Computer Model to calculate absolute and relative probabilities associated with each SP-multiple-alignment that it creates. Details of how probabilities are calculated, using values for the frequency of occurrence of SP-patterns, may be found in [63, Section 4.4] and [61, Section 3.7].

Because SP-multiple-alignments are the basis for all the several aspects of intelligence exhibited by the SP Computer Model—unsupervised learning, natural language processing, pattern recognition, several kinds of probabilistic reasoning, and more—the probabilities calculated for each SP-multiple-alignment provide probabilities as required for each aspect of intelligence. For example, with each instance

of a given kind of probabilistic reasoning, a measure of probability may serve as a measure of the level confidence one may have in that line of reasoning.

## 3.5   Unsupervised learning

Unsupervised learning in the SP System means processing one or more New SP-patterns to develop one or more collections of Old SP-patterns which, via the creation of SP-multiple-alignments, can encode the given set of New SP-patterns economically. Each such collection of Old SP-patterns is called an *SP-grammar*.

In this process of unsupervised learning, Old SP-patterns may be created directly from New SP-patterns, but most of them are likely to have been created via partial matches between New and Old SP-patterns. As with the building of SP-multiple-alignments, heuristic search with backtracking is normally needed to find SP-grammars that are 'good'.

The SP Computer Model has already demonstrated an ability to learn generative SP-grammars from unsegmented samples of English-like artificial languages, including segmental structures, classes of structure, and abstract patterns, and to do this in an 'unsupervised' manner ([63, Section 5], [61, Chapter 9]). But there are (at least) two shortcomings in the system [63, Section 3.3]: 1) it cannot learn intermediate levels of structure in an SP-grammar, and 2) it cannot learn discontinuous dependencies in such a grammar. These two shortcomings in learning apply, although the SP-multiple-alignment framework can accommodate structures of those kinds. It appears that those two problems may be overcome and that their solution would greatly enhance the capabilities of the SP Computer Model in unsupervised learning.

## 3.6   SP-Neural

Key concepts in the SP Theory may be mapped on to structures of neurons and their interconnections in a version of the SP Theory called *SP-Neural*. Current thinking about the structure and workings of SP-Neural, how it relates to known features of the brain, and how the concepts may be developed, is described in [68].

## 3.7   Strengths and potential of the SP System

Distinctive features and advantages of the SP System compared with other AI-related systems are described in [69].

The strengths and potential of the SP System are described quite fully in [63], and in much more detail in [61]. In brief, the SP System has strengths and potential in four main areas summarised here:

- *Versatility in aspects of intelligence* including: unsupervised learning; the analysis and production of natural language; pattern recognition that is robust in the face of errors; pattern recognition at multiple levels of abstraction; computer vision; best-match and semantic kinds of information retrieval; several kinds of reasoning (next item); planning; and problem solving. There is more detail in [63] and [61].

- *Versatility in reasoning* including: one-step 'deductive' reasoning; chains of reasoning; abductive reasoning; reasoning with probabilistic networks and trees; reasoning with 'rules'; nonmonotonic reasoning and reasoning with default values; Bayesian reasoning with 'explaining away'; causal reasoning; reasoning that is not supported by evidence; the inheritance of attributes in class hierarchies; and inheritance of contexts in part-whole hierarchies. There is more detail in [63, Section 10] and [61, Chapter 7]. There is also potential for spatial reasoning [65, Section IV-F.1] and what-if reasoning [65, Section IV-F.2].

- *Versatility in the representation of diverse kinds of knowledge* including: the syntax of natural languages; class-inclusion hierarchies (with or without cross classification); part-whole hierarchies; discrimination networks and trees; if-then rules; entity-relationship structures; relational tuples; and concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition. The addition of two-dimensional SP-patterns to the SP Computer Model is likely to expand the representational repertoire of the SP System to structures in two-dimensions and three-dimensions, and the representation of procedural knowledge with parallel processing. There is more detail in [66, Section III-B], and in [63, 61].

- *Seamless integration.* Because of the versatility of the SP System as outlined above, and because this versatility is largely due to the central role of SP-multiple-alignment, there is clear potential for the seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, in any combination. It appears that that kind of seamless integration is *essential* in any artificial system that aspires to the fluidity, versatility and adaptability of the human mind.

Figure 4 shows schematically how the SP System, with SP-multiple-alignment at centre stage, exhibits versatility in its capabilities and their seamless integration.

In view of the versatility of the SP System, and the seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, and since those AI-related strengths of the SP System are due largely to the versatility of the SP-multiple-alignment construct, there are reasons to believe that the concept
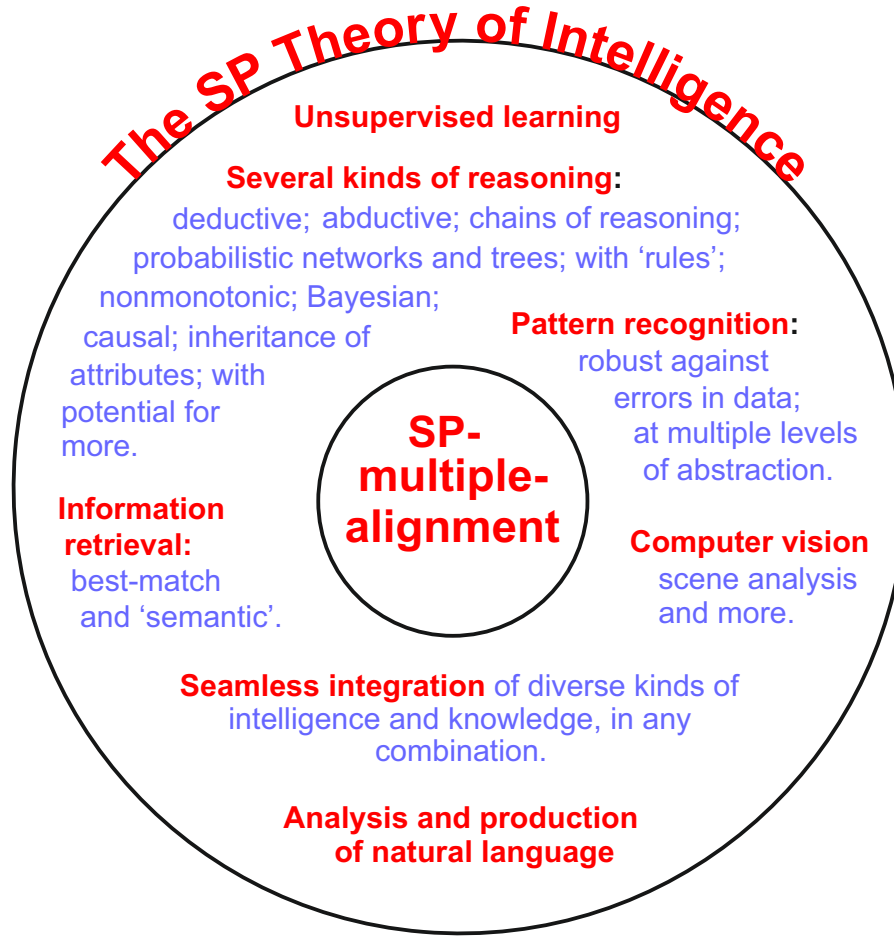
15

Figure 4: A schematic representation of versatility and integration in the SP System, with SP-multiple-alignment at centre stage. Adapted from Figure 6 in [71], with permission.

of SP-multiple-alignment may prove to be as significant for an understanding of human intelligence as is DNA for biological sciences: it may come to be seen as the 'double helix' of intelligence.

## 3.8 The SP Machine

It is envisaged that the SP Computer Model will provide the basis for the development of a highly-parallel *SP Machine*, as shown schematically in Figure 5. This projected development, described in [39], would be a vehicle for further research,

16

and ultimately the basis for a system with the scale and robustness needed for scientific, industrial, commercial, and administrative applications.
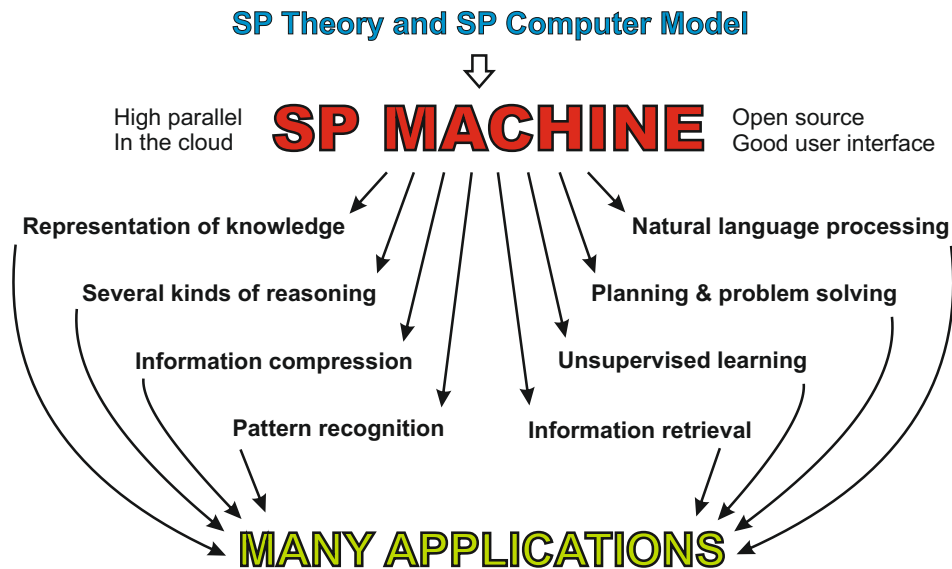


Figure 5: Schematic representation of the development and application of the SP machine. Reproduced from Figure 2 in [63], with permission.

## 3.9   Potential benefits and applications of the SP System

The SP System has potential in several areas of application including: helping to solve nine problems with big data; helping in the development of human-like intelligence in autonomous robots; helping in the understanding of human vision and in the development of computer vision; helping with medical diagnosis; functioning as a database system with intelligence; and more.

Details of peer-reviewed papers and other documents about the potential benefits and applications of the SP System may be found, with download links on bit.ly/2Gxici2.

As its title suggests, a paper called "Unsolved problems in AI, described in the book *Architects of Intelligence* by Martin Ford, and how they may be solved via the SP System" [72] describes how the SP System may solve at least 10 problems in AI, described by experts in AI in interviews with the science journalist, Martin Ford [20].

# 4 Related research

Research relating to the main thesis of this paper is considered in the subsections that follow.

## 4.1 Established techniques for IC

Established techniques for the compression of information such as Huffman coding, arithmetic coding, and wavelet compression, have a mathematical flavour (see, for example, [50]). Since techniques like those have a good pedigree and have proved their worth in many applications, one might suppose that they would be the starting point for any research, like the SP programme of research, where IC has a central role—and they would be the starting point for any discussion of how mathematics may be understood in terms of IC. But:

- The SP programme of research (Section 3) has adopted a different perspective. It attempts to reach down below the mathematics of other approaches, and, as noted in Section 3, it focusses on ICMUP, the relatively simple, 'primitive' idea that IC may be understood as a search for patterns that match each other, with the merging or 'unification' of patterns that are the same.

- Since ICMUP is a relatively 'concrete' idea, which is less abstract than much of mathematics, it suggests avenues that may be explored in understanding possible mechanisms for IC in artificial systems like the SP System, including SP-Neural, the 'neural' version of the SP System (Section 3.6 and [68]).

- Perhaps most importantly, in any discussion of the fundamentals of mathematics, it would not be appropriate for anything except peripheral arguments to use mathematics itself.

## 4.2 Algorithmic probability and algorithmic information theory

As with Huffman coding, arithmetic coding, and wavelet compression, it may seem that, because they relate closely to IC, two other concepts would have a bearing on the design and workings of the SP System. These are the *Algorithmic Probability Theory* (ALP, pioneered by Ray Solomonoff [53, 54], [32, Chapter 4]) and *Algorithmic Information Theory* (AIT, based on ALP and pioneered by Andrey Kolmogorov [31, 30] and Gregory Chaitin [15, 13, 12], see also [32, Chapter 2]). But, for reasons given in this subsection, their usefulness for present purposes is not as great as one may suppose.

### 4.2.1 The shortest computer program

Informally, the central idea in ALP and AIT is that the information content, or 'complexity', of any given string of atomic symbols is equivalent to the length (in bits) of the shortest computer program (for a 'universal Turing machine') that can create that string.

In simple cases, it may be possible to prove that the shortest program has been found, but normally one can only say that the 'shortest program' is the shortest that one or more people have been able to find or create after a certain amount of effort, perhaps with assistance from an established compression algorithm.

In ALP, the bit length of the shortest program is used, via Thomas Bayes' Theorem (see Section 8.1) to assign to objects an '*a priori* probability' that is in some sense universal. Marcus Hutter and colleagues [27] describe applications for ALP which include "Solomonoff induction" (Section 4.1), "expected time/space complexity of algorithms under the universal distribution" (Section 4.3), and "halting probability" (Section 4.5).

With AIT, the 'shortest program' idea has applications in three main areas described by Hutter [26]: philosophy (Section 6.1: helping to formalize and quantify such concepts as simplicity and complexity in the foundations of thermodynamics, and helping to solve the problem of Maxwell's demon), practice (Section 6.2: applications in linguistics and genetics, and in the development of a "universal similarity metric"), and science (Section 6.3: applications in mathematics, theoretical computer science, statistics, cognitive sciences, biology, physics, economics, and machine learning).

### 4.2.2 A different perspective

Although ALP and AIT are important, the SP programme of research has a different perspective:

- Unlike ALP and AIT, the SP System is *not* founded on the concept of a universal Turing machine. Instead:

  - A central idea in the SP System is the conjecture that *all* kinds of information processing may be achieved via IC.

  - More specifically, the SP System is dedicated to ICMUP.

  - And more specifically again, a central part of the SP System is the concept of SP-multiple-alignment (Section 3.3), which, as described in Section 5.7, is itself a generalisation of the first six variants of ICMUP described in Section 5.

- Although the details have not been fully worked out, it seems that the concept of a 'universal Turing machine', and the equivalent concept of a 'Post canonical system' [46], may themselves be seen as special cases of the SP System, as described in [61, Chapter 4].

- The SP System provides much of the human-like intelligence (Section 3.7) which, as Turing recognised [56, 58], is missing from the concept of a universal Turing machine.

- The Bayesian view of probability (which, as noted in Section 4.2.1, makes a contribution to the concept of ALP) is very different from the 'frequentist' view of probability which has been adopted in the SP System (Section 8.1).

In connection with these issues, an interesting example is provided by the decimal expansion of $\pi$ (3.14159265358979 ...) which has a small algorithmic complexity (because it can be created by a simple program) but which appears to most people to be entirely random. The fact that the SP System, like most people, fails to recognise the underlying regularity of this sequence may be seen as indirect evidence, in conjunction with other evidence, that ICMUP is a unifying principle in HLPC [71, Section 19].

## 4.3   Algorithmic cognition

Another area of research which aims to develop new insights into the nature of human cognition is *Algorithmic Cognition* (see, for example, [22, 73]). This perspective takes advantage of insights gained in the development of ALP (with Bayes' Theorem) and AIT (Section 4.2). It comes after that section on ALP and AIT to save having to repeat what is there.

Achievements with Algorithmic Cognition include:

"... we have offered what we think is an essential and what appears a necessary connection between the concept of cognition and algorithmic information theory. Indeed, within cognitive science, the study of working memory, probabilistic reasoning, the emergence of structure in language, strategic response, and navigational behavior is cutting-edge research. In all these areas we have made contributions [references given] based upon algorithmic complexity as a useful normative tool, shedding light on mechanisms of cognitive processes." [22, Section 4].

and

"In the cognitive sciences, the study of working memory, of probabilistic reasoning, the emergence of structure in language, strategic response,

and navigational behavior is cutting-edge research. In all these areas
the algorithmic approach to cognition has made contributions [refer-
ences given] based upon algorithmic probability as a useful normative
tool, shedding light on the algorithmic mechanisms of cognitive pro-
cesses." [73, Conclusion]

Algorithmic Cognition and the SP System are two different 'flavours' of research
on human cognition, each with distinctive contributions to make. In the interests
of a diversity in approaches to the solution of difficult problems, this is as it should
be.

## 4.4 Structuralism and mathematics-as-a-science-of-patterns

The 'structuralism' view of the foundations of mathematics, mentioned in Section
2.1 above, is considered a little more fully in this subsection.

### 4.4.1 Resnik

In the previously-mentioned *Mathematics as a Science of Patterns* by Resnik [47],
he describes his concept of 'structure' and makes clear that he is using that term
to mean essentially the same thing as 'pattern':

"... for some time the practice of pure mathematics has reflected the idea
that mathematics is concerned with structures involving mathematical
objects and not with the 'internal' nature of the objects themselves. ...

"The underlying philosophical idea here is that in mathematics the pri-
mary subject-matter is not the individual mathematical objects but
rather the structures in which they are arranged. The objects of math-
ematics, that is, the entities which our mathematical constants and
quantifiers denote, are themselves atoms, structureless points, or posi-
tions in structures. And as such they have no identity or distinguishing
features outside a structure.

"For epistemological purposes I find it more suggestive to speak of
mathematical patterns and their positions rather than of structures.
... In what follows I will use the terms 'pattern' and 'structure' more
or less interchangeably." ([47, Locations 2310–2316]).

This concept of pattern or structure has some similarity to the concept of 'SP-
pattern' in the SP System: an SP-pattern is "an array of atomic SP-symbols in

one or two dimensions, where an 'SP-symbol' is simply a distinctive mark that can be matched in a yes/no manner with any other SP-symbol." (Section 3).

But the concept of 'SP-pattern' in the SP System appears to differ from Resnik's concept of 'pattern' because each SP-symbol belongs to an alphabet of SP-symbols in which an SP-symbol representing one class of SP-symbols in the alphabet (e.g., 'A') can be distinguished from any SP-symbol representing any other class of SP-symbols in the alphabet ((e.g., 'B')), and so SP-symbols do have "identity or distinguishing features outside a structure".

### 4.4.2   Shapiro

In his book on the *Philosophy of Mathematics: Structure and Ontology* [52], Stewart Shapiro writes:

> "... pure mathematics is the study of structures, independently of whether they are exemplified in the physical realm, or in any realm for that matter. The mathematician is interested in the internal relations of the places of these structures, and the methodology of mathematics is, for the most part, deductive." [52, Locations 1139–1145], and later:

> "...the boundary between mathematics and ordinary discourse is at least as fuzzy as the boundary between mathematics and science." [52, Locations 4143–4147] and "... one result of the structuralist perspective is a healthy blurring of the distinction between tween mathematical and ordinary objects: ..." [52, Location 4147].

These points are discussed in Section 4.4.4, below.

### 4.4.3   Burgin

In his book on *Structural Reality* [11], Mark Burgin, after describing how structures may be seen in physics, writes:

> "Structures appear not only in physics. Personality has its structure, there is the structure of a novel, any language contains many structures and even a dream has its structure. As a result, in the context of the theory of named sets, it was discovered that everything in the world has a structure , even chaos (... [10]). Vacuum is considered as absence of matter, void, but physicists study the *structured vacuum* [2]. Thus structures exist not only in languages, society or human personality, but everywhere. Consequently, it is necessary to study structures not only in linguistics, psychology and anthropology, which were the first areas where the structural approach called *structuralism* was applied,

but in all sciences: in natural sciences, such as physics [and] social sciences, ... [in] humanities, such as sociology and psychology, and [in] technical sciences, such as computer science." [11] (emphasis in the original).

These points are discussed in Section 4.4.4, next.

### 4.4.4  Structuralism, IC, and the SP System

Structuralism relates to the SP System because the SP programme of research recognises that structures are pervasive in HLPC, and because the SP System is largely about the representation, learning, and application of HLPC structures.

An important difference between the SP System and structuralism as it has been developed by the authors quoted in Sections 4.4.1, 4.4.2, and 4.4.3, is that those authors appear to have no role for IC or ICMUP.

As an example, IC has proved to be a key to understanding how, in the (unsupervised) learning of their first language (or languages), young children may isolate words as significant structures or patterns in the language that they hear which, almost invariably, has no systematic physical markers to the beginnings and ends of words [71, Section 15.1]. There is evidence that similar principles apply to the unsupervised learning of the phrase structure of natural language [71, Section 15.2], and the unsupervised learning of other syntactic structures [71, Section 16].

The idea that IC may be the key to the learning or discovery of 'natural' structure in the world has been dubbed the 'DONSVIC' principle, short for 'The Discovery of Natural Structures Via Information Compression' [63, Section 5.2]. It appears to be relevant, not only to the discovery of linguistic structures as in the examples just given, but also to the discovery of three-dimensional structures as described in [64, Sections 6.1 and 6.2], and, by conjecture, with potential for the discovery of *any* kind of structure in the world.

The generality of the SP System in modelling structures in any domain sits well with: "... pure mathematics is the study of structures, independently of whether they are exemplified in the physical realm, or in any realm for that matter." (quoted in Section 4.4.2); and the "blurring of the distinction between mathematical and ordinary objects" (quoted in the same section); and the very wide applicability of structuralism as described by Burgin (in the quotation in Section 4.4.3).

## 5  Seven techniques for ICMUP

As its title suggests, this section describes seven techniques for ICMUP. They are fundamental in the SP System and are central in the main thesis of this paper,

that, to a large extent, mathematics may be understood as ICMUP.

While care has been taken in this programme of research to avoid unnecessary duplication of information across different publications, the importance of the following seven variants of ICMUP has made it necessary, for the sake of clarity, to describe them quite fully both in this paper and also in [71].

## 5.1 Basic ICMUP

The simplest of the techniques to be described is to find two or more patterns that match each other within a given body of information, **I**, and then merge or 'unify' them so that multiple instances are reduced to one. This is illustrated in the upper part of Figure 6 where two instances of the pattern 'INFORMATION' near the top of the figure has been reduced to one instance, shown just above the middle of the figure. Below it, there is the pattern 'INFORMATION', with 'w62' appended at the front, for reasons given in Section 5.2, below.

Here, and in subsections below, we shall assume that the single pattern which is the product of unification is placed in some kind of dictionary of patterns that is separate from **I**.

The version of ICMUP just described will be referred to as *basic ICMUP*.

A detail that should not distract us from the main idea is that, when compression of a body of information, **I**, is to be achieved via basic ICMUP, any repeating pattern that is to be unified should occur more often in **I** than one would expect by chance for a pattern of that size.

## 5.2 Chunking-with-codes

A point that has been glossed over in describing basic ICMUP is that, when a body of information, **I**, is to be compressed by unifying two or more instances of a pattern like 'INFORMATION', there is a loss of information about the *location* within **I** of each instance of the pattern 'INFORMATION'. In other words, basic ICMUP achieves 'lossy' compression of **I**.

This problem may be overcome with the *chunking-with-codes* variant of ICMUP:

- A unified pattern like 'INFORMATION', which is often referred to as a 'chunk' of information,[4] is stored in a dictionary of patterns, as mentioned in Section 5.1.
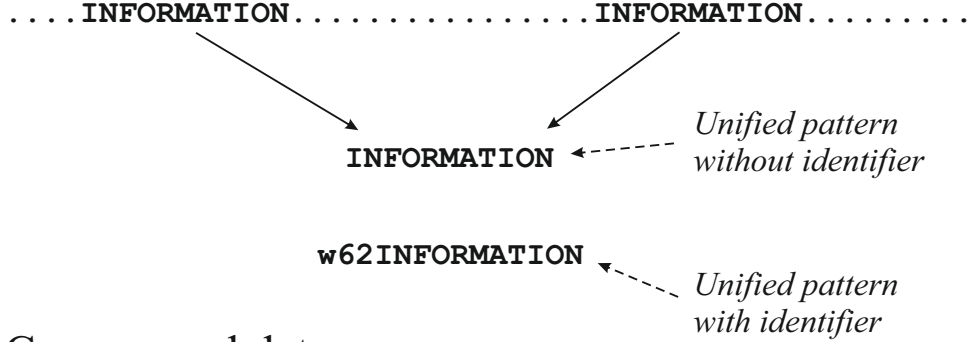
---

[4]There is a little more detail about the concept of 'chunk' in [71, Section 2.4.2].

- Now, the unified chunk is given a relatively short name, identifier, or 'code', like the 'w62' pattern appended at the front of the 'INFORMATION' pattern, shown below the middle of Figure 6.

- Then the 'w62' code is used as a shorthand which replaces the 'INFORMATION' chunk of information wherever it occurs within **I**. This is shown at the bottom of Figure 6.

- Since the code 'w62' is shorter than each instance of the pattern 'INFORMATION' which it replaces, the overall effect is to shorten **I**. But, unlike basic ICMUP, chunking-with-codes may achieve 'lossless' compression of **I** because the original information may be retrieved fully at any time.

- Details here are: 1) that compression can be optimised by giving shorter codes to chunks that occur frequently and longer codes to chunks that are rare. This may be done using some such scheme as Shannon-Fano-Elias coding, described in, for example, [17]; and 2) By ensuring that any chunk, **C**, to be given this treatment should be more frequent in **I** than the minimum needed (for a chunk of the size of **C**) to achieve compression (Section 5.1), and by ensuring that the size of every code is optimal, there should be an overall compression of **I**.

## Raw data

```
....INFORMATION................INFORMATION.........


        INFORMATION
```

*Unified pattern
without identifier*

```
     w62INFORMATION
```

*Unified pattern
with identifier*

## Compressed data

```
....w62...............w62.........
```

Figure 6: A schematic representation of the way two instances of the pattern 'INFORMATION' in a body of data may be unified to form a single 'unified pattern', shown just above the middle of the figure. To achieve lossless compression, the relatively short identifier 'w62' may be assigned to the unified pattern 'INFORMATION', as shown below the middle of the figure. At the bottom of the figure, the original data may be compressed by replacing each instance of 'INFORMATION' with a copy of the relatively short identifer, 'w62'. Adapted from Figure 2.3 in [61], with permission.

## 5.3   Schema-plus-correction

A variant of the chunking-with-codes version of ICMUP is called *schema-plus-correction*. Here, the 'schema' is like a chunk of information and, as with chunking-with-codes, there is a relatively short identifier or code that may be used to represent the chunk.

What is different about the schema-plus-correction idea is that the schema may be modified or 'corrected' in various ways on different occasions.

For example, a menu for a meal in a cafe or restaurant may be something like 'MN: ST MC PG', where 'MN' is the identifier or code for the menu, 'ST' is a variable that may take values representing different kinds of 'starter', 'MC' is a variable that may take values representing different kinds of 'main course', and 'PG' is a variable that may take values representing different kinds of 'pudding'.

With this scheme, a particular meal may be represented economically as something like 'MN: ST(st2) MC(mc5) PG(pg3)', where 'st2' is the code or identifier for 'minestrone soup', 'mc5' is the code for 'vegetable lassagne', and 'pg3' is the code

26

for 'ice cream'. Another meal may be represented economically as 'MN: ST(st6) MC(mc1) PG(pg4)', where 'st6' is the code or identifier for 'prawn cocktail', 'mc1' is the code for 'lamb shank', and 'pg4' is the code for 'apple crumble'; and so on. Here, the codes for different dishes serve as modifiers or 'corrections' to the categories 'ST', 'MC', and 'PG' within the schema 'MN: ST MC PG'.

## 5.4   Run-length coding

A third variant, *run-length coding*, may be used where there is a sequence of two or more copies of a pattern, each one except the first following immediately after its predecessor like this:

'INFORMATIONINFORMATIONINFORMATIONINFORMATIONINFORMATION'.

In this case, the multiple copies may be reduced to one, as before, something like 'INFORMATION×5', where '×5' shows how many repetitions there are; or something like '[INFORMATION*]', where '[' and ']' mark the beginning and end of the pattern, and where '*' signifies repetition (but without anything to say when the repetition stops).

In a similar way, a sports coach might specify exercises as something like "touch toes (×15), push-ups (×10), skipping (×30), ..." or "Start running on the spot when I say 'start' and keep going until I say 'stop'".

With the 'running' example, "start" marks the beginning of the sequence, "keep going" in the context of "running" means "keep repeating the process of putting one foot in front of the other, in the manner of running", and "stop" marks the end of the repeating process. It is clearly much more econonomical to say "keep going" than to constantly repeat the instruction to put one foot in front of the other.

## 5.5   Class-inclusion hierarchies

A widely-used idea in everyday thinking and elsewhere is the *class-inclusion hierarchy*: the grouping of entities into classes, and the grouping of classes into higher-level classes, and so on, through as many levels as are needed.

This idea may achieve ICMUP because, at each level in the hierarchy, attributes may be recorded which apply to that level and all levels below it—so economies may be achieved because, for example, it is not necessary to record that cats have fur, dogs have fur, rabbits have fur, and so on. It is only necessary to record that mammals have fur and ensure that all lower-level classes and entities can 'inherit' that attribute. In effect, multiple instances of the attribute 'fur' have been merged

or unified to create that attribute for mammals, thus achieving compression of information.[5]

This idea may be generalised to cross-classification, where any one entity or class may belong in one or more higher-level classes that do not have the relationship superclass/subclass, one with another. For example, a given person may belong in the classes 'woman' and 'doctor' although 'woman' is not a subclass of 'doctor' and *vice versa.*

## 5.6 Part-whole hierarchies

Another widely-used idea is the *part-whole hierarchy* in which a given entity or class of entities is divided into parts and sub-parts through as many levels as are needed. Here, ICMUP may be achieved because two or more parts of a class such as 'car' may share the overarching structure in which they all belong. So, for example, each wheel of a car, the doors of a car, the engine or a car, and so on, all belong in the same encompassing structure, 'car', and it is not necessary to repeat that enveloping structure for each individual part.

## 5.7 Generalisation of ICMUP via SP-multiple-alignment

The seventh version of ICMUP, the *SP-multiple-alignment* construct outlined in Section 3.3, encompasses all the preceding six versions of ICMUP.

How the preceding six versions of ICMUP may be modelled within the SP-multiple-alignment framework is described in [70, Section 2].

The strengths and potential of the SP-multiple-alignment construct in modelling aspects of human intelligence and the representation of knowledge is summarised in Section 3.7, described fairly fully in [63], and described in much more detail in [61].

# 6  Mathematics as ICMUP

This section presents the main MICMUP thesis of this paper: that much of mathematics, perhaps all of it, may be seen as ICMUP.

At this point it is perhaps appropriate to mention that there are two apparent contradictions of that idea: 1) With mathematics (and computing) it is easy to create large amounts of redundancy, which is the opposite of IC; 2) Redundancy

---

[5]The concept of class-inclusion hierarchies with inheritance of attributes is quite fully developed in object-oriented programming, which originated with the Simula programming language [9] and is now widely adopted in modern programming languages.

is often useful as, for example, in safeguarding information against loss or corruption. Those two apparent contradictions of the main thesis of this paper, and how they may be resolved, are discussed briefly in Appendix A, with references to [71, Appendix C], where fuller discussions may be found.

Since the arguments in Appendix A.1 depend on the arguments in this section, it is probably best to read it after reading this section.

## 6.1 An example of IC via mathematics

This subsection begins with an example showing how ordinary mathematics, without any specialised technique, can be very effective in compressing information.

The equation $s = (gt^2)/2$, which expresses one aspect of one of the laws of motion, is a very compact means of representing any table, including large ones, showing the distance, $s$, travelled by a falling object in a given time, $t$, since it started to fall, as illustrated in Table 1.[6] The constant, $g$, is the acceleration due to gravity—about $9.8 m/s^2$.

| Distance (m) | Time (sec) |
|---:|---:|
| 0.0 | 0 |
| 4.9 | 1 |
| 19.6 | 2 |
| 44.1 | 3 |
| 78.5 | 4 |
| 122.6 | 5 |
| 176.5 | 6 |
| 240.3 | 7 |
| 313.8 | 8 |
| 397.2 | 9 |
| 490.3 | 10 |
| 593.3 | 11 |
| 706.1 | 12 |
| 828.7 | 13 |
| 961.1 | 14 |
| 1103.2 | 15 |
| 1255.3 | 16 |
| Etc | Etc |

Table 1: The distance (in metres) travelled by a falling object in a given time (in seconds) since it started to fall.

---

[6]Of course, the law does not work for something like a feather falling in air.

That small equation would represent the values in the table even if it was a 1000 times or a million times bigger, and so on. Likewise for other equations such as $a^2 + b^2 = c^2$, $PV = k$, $F = q(E + v \times B)$, and so on.

To make these points, it is not strictly necessary to show Table 1. But the table helps to emphasise the contrast between the potentially huge volumes of data in such a table and the small size of the equation which describes those data—and, correspondingly, the potentially high levels of IC that may be achieved with ordinary mathematics which is not specialised for compression of information.

## 6.2 How ICMUP may be seen in the structures and workings of mathematics

The subsections that follow describe how some of the basic principles and techniques for the compression of information that were described in Section 5 may be seen in the structures and workings of mathematics.

In themselves, these examples do not prove that mathematics may be understood as being entirely devoted to the compression of information. But there are reasons to think that compression of information is fundamental in mathematics:

- Since the techniques to be described are techniques which are widely used in more complex forms of mathematics, it seems likely that mathematics may indeed be understood in its entirety as ICMUP.

- As described in Section 7.1.1, the workings of simple logical functions, including the NAND logical function, may be understood in terms of ICMUP. Since it is widely accepted that, in principle, the computational heart of any general-purpose digital computer may be constructed entirely from NAND gates [38], it appears that, within the bounds imposed by computational complexity, ICMUP has the generality to support any kind of computation, including mathematical computations.

## 6.3 Basic ICMUP

The simplest version of ICMUP, which may be called "basic ICMUP" (Section 5.1), may be seen in mathematics whenever one identifier is matched with another, with implicit unification of the two.

### 6.3.1 The matching and unification of identifiers: assigning a value to a variable

In mathematics, ICMUP may be seen wherever there is a need to invoke a named entity. If, for example, we want to calculate the value of $z$ from these three

equations: $x = 4$; $y = 5$; $z = x + y$, we need to match the identifier $x$ in the third equation with the identifier $x$ in the first equation, and to unify the two, so that the correct value is used for the calculation of $z$. Likewise for $y$.

For anyone familiar with computer programming, what has been described may seem simple enough. But in computer programs, a variable is more complex than its name. It has a structure that allows it to hold a 'value' and there are procedures for assigning values to variables. But none of that complexity should obscure the basic processes of matching and unification of identifiers, as described above.

### 6.3.2  The matching and unification of identifiers: calling a function

In a similar way if we wish to invoke or 'call' a function such as '$\log x$' (the logarithm of a number), there must be a match between the name of the function in the call to the function (such as '$\log 1000$') and the name of the function in its definition, '$\log x$'. Unification of the call to the function with the definition of the function may be seen to have the effect of assigning the number in the call (1000 in this example) to the variable $x$ in the definition of the function.

As before, the complexity of assigning a number to a variable should not obscure the simplicity of matching identifiers and unifying them.

### 6.3.3  The execution of a function

At an abstract level, any function may be seen as a table in which each row shows the connection between one or more input values and one or more output values. And simple functions, such as a one-bit adder, may be specified in exactly that way, as shown in Table 2.

| Input (1) | Input (2) | Sum | Carry |
|---:|---:|---:|---:|
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 |

Table 2: A table to define a function for the addition of two one-bit numbers in binary arithmetic, with provision for the carrying out of one bit.

To see how ICMUP features in the workings of this function, consider how the function would calculate a sum for the input values '1' and '0'. In this case: there is a search for a match between the first of those two input values (1) and the four values that appear in the first column of the table, leading to positive matches in the first two rows of the table; and there is a similar search for a match between

the second of those two input values (0) and the four values that appear in the second column of the table, leading to positive matches in the second and fourth rows of the table.

But the matches which, via unification, achieve the greatest overall compression (both '1' and '0' in the first and second columns in one row) have the effect of selecting the second row in the table. The sum obtained in this case is '1' (in the third column, second row), with the carry digit, '0' (in the fourth column, second row). Those two values are of course the correct result for the addition of the input values '1' and '0'.

### 6.3.4 Matching and unification of patterns with Peano's axiom for natural numbers

The sixth of Peano's axioms for natural numbers—for every natural number $n$, $S(n)$ is a natural number—provides the basis for a succession of numbers: $S(0)$, $S(S(0))$, $S(S(S(0)))$ ..., itself equivalent to unary numbers in which $1 = /$, $2 = //$, $3 = ///$, and so on. Here, $S$ at one level in the recursive definition is repeatedly matched and unified with $S$ at the next level.

## 6.4 Chunking-with-codes

This subsection describes aspects mathematics that may be seen to exemplify the chunking-with-codes technique for IC, as described in Section 5.2.

### 6.4.1 Named functions

If a body of mathematics is repeated in two or more parts of something larger then it is natural to declare it once as a named 'function', where the body of the function may be seen as a 'chunk' of information, and the name of the function is its 'code' or identifier. This avoids the need to repeat the body of the function in two or more places.

An example of this kind of thing is the calculations needed to find the square root of a number, often provided in spreadsheets, programming languages, and the like, as a ready-made square-root function with a name like '$\sqrt{x}$'. That name may be used to invoke the function wherever it is needed, like this: '$\sqrt{16}$'. Similar things may be done with functions such as '$\sin(x)$', '$\cos(x)$', and '$\log(x)$'.

Although they are not commonly seen as 'functions', all of the operations of addition, subtraction, multiplication, the power notation, and division, may be cast in that mould as, for example, '$plus(x,y)$', '$subtract(x,y)$', and so on. As such, they may be seen as examples of the chunking-with-codes device for compression of information. As we shall see in Section 6.5, they may also be seen as examples

of the schema-plus-correction device, and in Section 6.6, they provide examples of run-length coding.

### 6.4.2 The number system

Number systems with bases greater than 1, like the binary, octal, decimal and hexadecimal number systems, may all be seen to illustrate the chunking-with-codes technique for compressing information. For example:

- A unary number like '///////' may be referred to more briefly in the decimal system as '7'. Here, '///////' is the chunk and '7' is the code.

- A unary number like '/////////////////' may be split into two parts: '//////////' and '///////'. Then, in the decimal system, the first part, which is the number of 10s, would be represented by '1', and the second part, which is number of units, would be '7', giving us the decimal number '17'.

- Of course, this 'positional' system can be extended so that a digit in the third position from the right represents the number of 100s, a digit in the fourth position from the right represents the number of 1000s, and so on.

Here, we can see how the chunking-with-codes technique allows us to eliminate the repetition or redundancy that exists in all unary numbers except '/'. This means that large numbers, like 2035723, may be expressed in a form that is very much more compact than the equivalent unary number.

## 6.5 Schema-plus-correction

Most functions in mathematics, like those mentioned above, are not only examples of chunking-with-codes: they are also examples of the schema-plus-correction device for compressing information. This is because they normally require input via one or more 'arguments' or 'parameters'. For example, the square root function needs a number like 49 for it to work on. Without that number, the function is a very general 'schema' for solving square root problems. With a number like 49, which may be regarded as a 'correction' to the schema, the function becomes focussed much more narrowly on finding the square root of 49.

## 6.6 Run-length coding

Run-length coding appears in various forms in mathematics, often combined with other things. The key idea is that some entity, pattern, or operation is repeated two or more times in an unbroken sequence. Here are some examples:

- Since all numbers with bases above 1 may be seen to be compressed representations of unary numbers (Section 6.4.2), unary numbers may be regarded as more fundamental than non-unary numbers. If that is accepted, then, for example, '3 + 7' may be seen as a shorthand for the repeated process of transferring one unary digit from a group of seven unary digits to a group of three unary digits. Thus the expression '+7' within '3 + 7' may be seen as an example of run-length coding.

  Subtraction may be interpreted in a similar way when a smaller number is subtracted from a larger number.

- Multiplication is repeated addition. So, for example, '3 × 10' is the 10-fold repetition of the operation '$x + 3$', where '$x$' starts with the value '0'. Thus '×10' within '3 × 10' may be seen as run-length coding. Since addition may itself be seen as a form of run-length coding (as described in the preceding bullet point), multiplication may be seen as run-length coding on two levels.

- Division of a larger number by a smaller one (eg, '12/3') is repeated subtraction which, as with multiplication, may be seen as run-length coding. Of course there will be a 'remainder' if the larger number is not an exact multiple of the smaller number. As with addition as a part of multiplication, and addition as itself an example of run-length coding, subtraction as a part of division, and subtraction as run-length coding, means that division may be seen as run-length coding on two levels.

- The power notation (eg, '$10^9$') is repeated multiplication, and is thus another example of run-length coding. Since multiplication, as repeated addition, is a form of run-length coding, and since addition may be seen as run-length coding (the first bullet point above), the power notation may be seen as run-length coding on three levels!

- A factorial (eg, '25!') is repeated multiplication and subtraction.

- The bounded summation notation (eg, '$\sum_{i=1}^{5} \frac{1}{i}$') and the bounded power notation (eg, '$\prod_{n=1}^{10} \frac{n}{n-1}$') are shorthands for repeated addition and repeated multiplication, respectively. In both cases, there is normally a change in the value of one or more variables on each iteration, so these notations may be seen as a combination of run-length coding and schema-plus-correction.

## 6.7   Class-inclusion hierarchies

Classes and subclasses (Section 5.5) feature in mathematics as 'sets', both as a sometimes-disputed foundation for mathematics, and as a branch of mathematics.

The notion of 'inheritance' does not have the prominence in set theory that it does in object-oriented programming, but, nevertheless, ICMUP may be seen in other concepts associated with sets, described in Section 7.1.

## 6.8   Part-whole hierarchies

It seems that part-whole hierarchies are not much used in mathematics, except perhaps in set theory, but, as we shall see in Section 7.2, they are quite prominent in the mathematics-related discipline of computing.

## 6.9   SP-multiple-alignment

Preliminary work described in [61, Chapter 10] shows that the SP System, with SP-multiple-alignment centre-stage, has potential to model mathematical constructs and mathematical processes. This should not be altogether surprising since, as noted in Section 5.7, SP-multiple-alignments can do everything that can be done with the six variants of ICMUP described in Sections 5.1 to 5.6, and it provides for their seamless integration too.

Other reasons for believing that the SP System has potential to model many and perhaps all concepts and processes in mathematics are:

- The generality of IC as a means of representing knowledge in a succinct manner (Section 3.7).

- The central role of IC in the SP-multiple-alignment framework (Section 3.3).

- The versatility of the SP-multiple-alignment framework in aspects of intelligence and the representation of knowledge (Section 3.7).

- The close connection that is known to exist between IC and concepts of probability (Section 8).

## 6.10   Some equations

It seems that most equations that have become established in mathematics and science may be interpreted in terms of some combination of the techniques for compressing information described in Section 5. Thus:

- Einstein's equation, $E = mc^2$, illustrates run-length coding in its power notation ($c^2$) and in the multiplication of $m$ with $c^2$.

- Newton's equation, $s = (gt^2)/2$, that featured in Section 6.1, illustrates run-length coding in its power notation ($t^2$), in the multiplication of $g$ with $t^2$, and in the division of $(gt^2)$ by 2.

- Pythagoras's equation, $a^2 + b^2 = c^2$, illustrates run-length coding via the power notation in $a^2$, $b^2$, and $c^2$, and via the addition of $b^2$ to $a^2$ (the first bullet point in Section 6.6).

- Boyle's law, $PV = k$, illustrates run-length coding in the multiplication of $P$ by $V$.

- The charged particle equation, $F = q(E + v \times B)$, illustrates run-length coding in the multiplication of $v$ by $B$, in the multiplication of $(E + v \times B)$ by $q$, and in the addition of $v \times B$ to $E$.

- One of special relativity's equations for time dilation, $\Delta t' = \Delta t / \sqrt{1 - v^2/c^2}$, illustrates chunking-with-codes and schema-plus-correction in its use of the square root function, and it illustrates run-length coding in the division of $v^2$ by $c^2$, in the subtraction of $v^2/c^2$ from 1, and in the division of $\Delta t$ by $\sqrt{1 - v^2/c^2}$.

- In its use of bounded summation ($\sum$), Shannon's equation for entropy, $H = -\sum_i p_i \log_2(p_i)$, illustrates a combination of run-length coding and schema-plus-correction (as noted in Section 6.6). It also illustrates chunking-with-codes in its use of the $\log_2$ notation.

Since addition, subtraction, multiplication, the power notation, and division, may each be seen as an example of chunking-with-codes and schema-plus-correction (Sections 6.4 and 6.5), as well as run-length coding (Section 6.6), the same can be said about the appearance of those notations in each of the examples above.

# 7    Mathematics-related disciplines as ICMUP

It seems that, to a large extent, what has been said about mathematics in Section 6 also applies to the mathematically-related disciplines of logic and computing.[7] The following two subsections present some examples in support of that idea.

---

[7] Where computing has its modern sense of computation by machine.

## 7.1 Logic

Subsections that follow describe some evidence for ICMUP in logic and related disciplines.

As a preliminary, we may guess that logic may, to a large extent, be understood in terms that are much like MICMUP because of Frege's considerable and largely successful efforts in demonstrating that mathematics may be largely understood in terms of logic [33, Chapter 2]. In the light of his research, it seems likely that, if mathematics can be understood largely in terms of ICMUP, the same would be true of logic.

### 7.1.1 XOR and other logical operations

The XOR logical function, shown in Table 3, and other simple logical functions, may be defined and interpreted in much the same way as the one-bit adder shown in Table 2.

| Input (1) | Input (2) | Output |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |

Table 3: A table to define the XOR logical function.

As with the one-bit adder, the operation of the XOR function may be understood in terms of basic ICMUP. Input values such as 1 (first) and 0 (second) may be matched and unified with values in the corresponding 'input' columns of the table. With those two input values, the third row is selected because it yields most matches—which, with unification, also means the greatest compression of information. And of course the third row yields the correct output value, which in this example is 1.

There are two points of interest here:

- *The XOR Function and Artificial Neural Networks.* As is well known, Marvin Minsky and Seymour Papert [35] demonstrated that basic perceptrons of the kind that were available in the late 1960s could not produce correct results with the XOR function, a demonstration which, for a time, led to a fall in interest in artificial neural networks.

- *The Generality of the NAND Logical Function.* As noted in Section 6.2, the fact that the NAND logical function may, like XOR and other simple logical

functions, be understood in terms of ICMUP, and the generally-accepted idea that the computational heart of any general-purpose computer may, in principle, be constructed entirely from NAND gates [38], provide evidence in support of the idea that compression of information is fundamental in all kinds of computation including mathematical computations.

### 7.1.2 Deriving a sets from a multiset

In logic and mathematics, a 'multiset' or 'bag' is like a set but any element within the multiset may be repeated as, for example, in the multiset {a, b, a, c, b, b, c, a, c}.

Conversion of any such multiset into the corresponding set means matching each element within the multiset with every other element and, wherever a match is found, unifying the two elements, including elements that are the result of earlier unifications, thus achieving ICMUP. In this case, the multiset {a, b, a, c, b, b, c, a, c} is reduced to the set {a, b, c}.

### 7.1.3 The union and intersection of sets

In much the same way that a set may be derived from a multiset (Section 7.1.2), the union and intersection of two sets may be found by the matching and unification of elements, yielding a reduction in the overall size of the two sets when unification has been achieved. Thus, for example, the union of the sets {b, f, d, a, c, e} and {e, g, i, f, d, h} is {a, b, c, d, e, f, g, h, i}, with the intersection {d, e, f}. In accordance with ICMUP, the union is smaller than the two sets from which it was derived.

### 7.1.4 ICMUP in Prolog

Further evidence for the significance of ICMUP in logic is that systems like Prolog—a computer-based version of logic—may be seen to function largely via the matching and merging of patterns.

Here, the meaning of 'unification' in Prolog—comparing two terms to see if they can be made to represent the same structure—is quite close to the meaning of 'unification' in this paper.

### 7.1.5 Versatility in reasoning with the SP System

As noted in Section 3.7, the SP Computer Model demonstrates several kinds of reasoning including one-step 'deductive' reasoning, chains of reasoning, abductive reasoning, and more.

Because of the probabilistic nature of the SP System, these forms of reasoning are probabilistic. So it may seem that they have little to do with logic, because of its all-or-nothing character. But if it is accepted that logic, like mathematics, is probabilistic at a deep level—for reasons given in Section 8—then the versatility of the SP System in probabilistic reasoning may be seen as further evidence for the importance of ICMUP in logic.

## 7.2 Computing

As with logic, it seems likely that, since computing is closely related to mathematics, it may, like mathematics, be understood in terms of ICMUP. Evidence in support of that view is presented in subsections that follow.

### 7.2.1 Matching and unification of patterns in definitions of 'computing'

Emil Post's [46] "Canonical System", which is recognised as a definition of 'computing' that is equivalent to a universal Turing machine, may be seen to work largely via ICMUP [61, Chapter 4].

Much the same is true of the workings of the transition function in a universal Turing machine. This is essentially a look-up table like that shown in Table 4.

| Input (1) | Input (2) | Output (1) | Output (2) |
|---|---|---|---|
| $s_0$ | 1 | $s_0$ | $\gg$ |
| $s_0$ | 0 | $s_1$ | 1 |
| $s_1$ | 1 | $s_1$ | $\ll$ |
| $s_1$ | 0 | $s_2$ | $\gg$ |

Table 4: An example of a transition function in a universal Turing machine, represented as a look-up table, as described in the text. *Key:* '$\gg$' means "move the read/write head one place to the right"; '$\ll$' means "move the read/write head one place to the left". Based on the example in [4, Section 2], with permission.

Much as with the examples described in Sections 6.3.3 and 7.1.1, ICMUP may be seen, for example, in the matching and unification of input values '$s_1$' and '1' with corresponding values in the input columns of the table. In this case, the effect will be to select the third row in the table, with the output values '$s_1$' and '$\ll$'—which mean "Set the state of the machine to '$s_1$' and move the read/write head of the machine one place to the left".

In a similar way, ICMUP may be seen in the workings of the NAND logical function which, as noted in Sections 6.2 and 7.1.1 may in principle provide the computational heart of any general-purpose digital computer.

### 7.2.2 Some other examples of ICMUP in computing

Here, in brief, are some other putative examples of ICMUP in computing:

- *Basic ICMUP.* As in mathematics (Section 6.3), basic ICMUP may be seen in computing in the matching of identifiers for variables and in calls to functions.

- *Chunking-With-Codes and Schema-Plus-Correction.* As in Section 6.4, named functions in computing may be seen as examples of the chunking-with-codes version of ICMUP, and as in Section 6.5, functions with parameters may be seen as examples of the schema-plus-correction version of ICMUP.

- *Run-Length Coding.* As in mathematics (Section 6.6), run-length coding may be seen in computing in the basic arithmetic functions. It may also be seen in iteration statements like *while ...*, *do ... while ...*, *for ...*, or *repeat ... until ....* And it may be seen in the use of recursion in functions such as *factorial(x)* for the calculations of the factorial of any number.

- *Class-Inclusion Hierarchies and Part-Whole Hierarchies.* In computing, the creation of classes and hierarchies of classes is supported in such object-oriented programming languages as Simula, Smalltalk, C++, and many more. Part-whole hierarchies are also prominent in software. In both cases, ICMUP has a role to play, much as described in Sections 5.5 and 5.6.

- *Retrieving Data From Computer Memory.* It is true that electronic circuits provide the mechanism for finding an address in computer memory but, at a more abstract level, the process may be seen as one of searching for a match between the address held in the CPU and the corresponding address in computer memory. When a match has been found between the address in the CPU and the corresponding address in memory, there is implicit unification of the two.

- *Query-by-Example.* Query-by-example, which is a popular technique for retrieving information from databases, may be seen to be essentially a process of finding good matches between a query pattern and patterns in the database, with unification of the best matches.

## 8 The intimate relation between IC and concepts of probability

The main focus of this paper is on MICMUP, but in view of the previously-noted very close relation between IC and concepts of probability, and because of the

importance of probabilities in theories of AI and HLPC, it is relevant to explore that relationship in a little more detail.

The relationship is evident in classical *Information Theory* [51], but has been explored most notably in Ray Solomonoff's *Algorithmic Probability Theory* (ALP, [53, 54], and [32, Chapter 4]).

In the SP System, the very close connection between IC and concepts of probability (and inference) makes sense in terms of ICMUP because:

- *IC via the unification of SP-patterns.* The unification of two or more copies of an SP-pattern achieves compression of information.

- *Frequencies of occurrence.* For a given SP-pattern, its frequency of occurrence may be derived via ICMUP from the number of original SP-patterns that have been unified to create that SP-pattern. Likewise for the frequency of occurrence of any SP-symbol.

- *Deriving absolute and relative probabilities from frequencies of occurrence.* In general, absolute and relative probabilities in the SP System may be derived from the frequencies of occurrence of SP-patterns ([63, Section 4.4], [61, Section 3.7]).

- *Inference and conditional probability.* Inference may be achieved via partial matching as, for example, in the way that seeing black clouds allows us to make the inference that rain is likely, via a partial match between '`black clouds`' and the pre-established SP-pattern '`black clouds rain`'. This is sometimes called *prediction by partial matching* (see, for example, [55]).

  The justification for this kind of inductive reasoning is itself the subject of much debate. There is a contribution to that debate in Appendix B.

Thus, the prominence of ICMUP in mathematics, and logic and computing, as described in this paper, suggests that some aspects of mathematics, and perhaps some aspects of logic and computing, are somehow probabilistic. But that seems to conflict with the familiar, all-or-nothing, clockwork nature of much of mathematics and logic, where $2 + 2 = 4$ and where Socrates is mortal (because he is human, and all humans are mortal), without uncertainties of any kind.

This apparent contradiction may, perhaps, be resolved in some such manner as the following:

- *Uncertainties associated with Gödel's incompleteness theorems.* Gregory Chaitin writes:

"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." [14, p. 80].

As indicated in this quotation, randomness in number theory is closely related to Gödel's incompleteness theorems (see, for example, [36]). Gödel's theorems depend on the phenomenon of recursion, a feature of many formal systems, several of Escher's pictures, and much of Bach's music, as described in some detail by Douglas Hofstadter in *Gödel, Escher, Bach: An Eternal Golden Braid* [24].

If "God sometimes plays dice with whole numbers" as suggested by Chaitin in the quote above, then at least one part of number theory is probabilistic. But it seems reasonable to assume that other parts of number theory (and thus mathematics), which probably include the parts where $2 + 2 = 4$, are *not* probabilistic. And we may guess that similar things apply to logic and computing.

- *Uncertainties associated with frequencies.* By contrast with the kinds of uncertainties just described, there are other uncertainties in the SP System, and associated probabilities, arising from the values for frequency of occurrence of SP-patterns and SP-symbols. These may serve in the calculations of probabilities that are made by the SP Computer Model (Section 3.4).

  But although the SP System is well suited to probabilistic applications, it seems likely that, with appropriate data, there will be other kinds of application where all values for probability are 0 or 1, so that the SP System will behave in the all-or-thing clockwork manner of conventional computing.

Although there is a very close relation between IC and concepts of probability, this does not mean that they are equivalent or inter-changeable. It appears that, in research in AI, cognitive science, and concepts of probability, there are advantages in putting one's main focus on IC. Arguments for that view are summarised in Section 8.2, below.

## 8.1 Classical, Frequentist, and Bayesian views of probability

Perhaps because of the prominence of uncertainties in the way people perceive things and think, much research in AI and cognitive science is based on concepts of probability such as those in Bayes' Theorem.

In brief, Bayes' Theorem may summarized with the equation $P(h|D) = \frac{P(D|h)P(h)}{P(D)}$, where $P(h)$ = prior probability of hypothesis $h$, $P(D)$ = prior probability of training data $D$, $P(h|D)$ = probability of $h$ given $D$, and $P(D|h)$ = probability of $D$ given $h$.

To be clear, the concept of probability in the SP System (Section 3.4) differs sharply from the concept of probability in Bayesian theory because it depends directly on the frequencies of occurrence of entities or events—derived from frequencies of occurrence of SP-patterns and SP-symbols—whereas Bayesian theory adopts the view that probability is a *degree of belief* in the probability of an entity or event, and includes the concept of *prior probability*, meaning the degree of belief in the probability of an entity or event prior to the receipt of new data.

Of course, the SP Theory, like any theory of probabilities that emphasises the importance of frequencies of occurrence of entities or events, may recognise probabilities that are prior to the receipt of new data, but such probabilities do not have the kind of special theoretical status that they do in Bayesian theory.

## 8.2 Asymmetry in the relationships between ICMUP and concepts of probability

The very close connection between IC and concepts of probability in both ordinary computing and in the SP System may suggest that there is nothing to choose between IC and concepts of probability as a foundation for theorising in AI, in cognitive science, and in research about probability. But for reasons outlined in the following subsections, *there are advantages in approaching probability via ICMUP—in AI, in cognitive science, and in other areas where values for probability are needed.*

### 8.2.1 Loss of information in the derivation of probabilities

As may be seen from points made above, absolute and conditional probabilities may be derived via ICMUP, but the reverse is not true. This is partly because, arguably, the matching and unification of patterns is more primitive than concepts of probability. But more to the point, values for probability, in themselves, have lost information about the matches and unifications that led to their creation.

Because probabilities may be derived from ICMUP but not the other way round, and because ICMUP is prominent in HLPC [71], *any artificial system that aspires to the generality of human intelligence should be founded on ICMUP, not concepts of probability.* In a similar way, it seems appropriate that ICMUP should be the basis for probability in theories of human cognition, and other areas where values for probability are needed.

### 8.2.2 Potential for creating new structures via ICMUP

Much can be done with Bayesian and other probabilistic approach to AI, but something is missing: it is assumed that all of the conceptual entities in a probabilistic analysis have been created already, and there is nothing about how they may be formed. By contrast, ICMUP in the SP System opens up the possibility of isolating words as discrete entities in speech [71, Section 15.1], and likewise for phrases [71, Section 15.2]. And it can provide a basis for the building of three-dimensional models of entities, as outlined in [64, Sections 6.1 and 6.2].

### 8.2.3 The scope of frequency information may be extended via ICMUP

It is often assumed that, when the frequency of occurrence of entities or events is used as the basis of probability measures, high frequencies are needed to ensure that results are statistically significant. Thus, for example:

> "There is a definition of probability in terms of frequency that is sometimes usable. It tells us that a good estimate of the probability of an event is the frequency with which it has occurred in the past. This simple definition is fine in many situations, but breaks down when we need it most; i.e., *its precision decreases markedly as the number of events in the past (the 'sample size') decreases. For sample sizes of 1 or 2 or none, the method is essentially useless.*" ([54, pp. 74–75], emphasis added).

By contrast, in searching for repeating patterns that may be unified to yield compression of information, *the sizes of repeating patterns are as important as their frequency.* Maximising the amount of redundancy found means maximising $R$ where:

$$R = \sum_{i=1}^{i=n} (f_i - 1) \cdot s_i, \tag{1}$$

and where $f_i$ is the frequency of the $i$th member of a set of $n$ patterns and $s$ is its size in bits. In brief, patterns that are both big and frequent are best.

But *there is a trade-off between the sizes of patterns and the minimum frequency that is needed for IC.* With small patterns, high frequencies are required. But with large patterns, useful compression can be achieved with frequencies that are as low as 2 or 3 [61, Sections 2.2.8.3 and 2.2.8.4].

As an example, a song (or other piece of music) that we have heard only once may be recognised from hearing, only once, a smallish sample of that song. Accordingly, we assign a mental probability of 1.0 to the identification we have made,

a probability which corresponds to a frequency of 2, because the first learning-by-hearing of the song yields a frequency of 1, and the second recognition-by-hearing of the song yields a frequency of 2. Of course, these arguments do not apply if either or both of the song, or the later-heard sample from the song, are very short.

### 8.2.4 Probability, causation, and structure

Another apparent shortcoming of relying too much on concepts of probability arises if we wish to know about causation. For example:

> "The answer [to difficulties in solving causal problems with statistics ...] has to do with the official language of statistics—namely the language of probability. This may come as a surprise to some [people] but the word *cause* is not in the vocabulary of probability theory; we cannot express in the language of probabilities the sentence *mud does not cause rain*—all we can say is that the two are mutually correlated or dependent—meaning that if we find one, we can expect the other. Naturally, if we lack a language to express a certain concept explicitly, we can't expect to develop scientific activity around that concept." [43, p. 342].

> "[An engineering diagram] is, in fact, one of the greatest marvels of science. It is capable of conveying more information than millions of algebraic equations or probability functions or logical expressions. What makes [such a diagram] so much more powerful is the ability to predict not merely how the [system] behaves under normal conditions but also how [it] will behave under millions of *abnormal* conditions." [43, p. 344].

To summarise what Pearl says here: *any satisfactory account of causation requires a description of relevant structures, not merely numbers and equations.*

Although no serious attempt has yet been made to examine issues in causality in terms of the SP Theory (but see [63, Section 10.5] and [61, Section 7.9]), there are reasons to think that it may be more successful than classical statistics. This is because, in accordance with the second quote above, the SP System has the potential to represent and to learn the kinds of structures that are needed for a comprehensive causal analysis (Section 8.2.2).

## 9 So what?

While it may be accepted that mathematics may be understood as ICMUP, readers may wonder what advantages, if any, may be derived from MICMUP and related

ideas. Here, in the following three subsections, are three possibilities.

## 9.1 An improved ratio of Simplicity to Power for ICMUP and the SP System

The discovery that the explanatory range of ICMUP may, without any increase in complexity, be extended from AI-related areas to mathematics and the mathematics-related areas of logic and computing, can mean an improvement in its ratio of Simplicity to descriptive and explanatory Power.

Since SP-multiple-alignment may be seen to be a generalisation of six variants of ICMUP (Section 5.7), and since it is a large part of the SP System (Section 3.3), the improved ratio of Simplicity to Power for ICMUP means a similar improvement for the SP System.

## 9.2 Potential for the development of a *New Mathematics*

In view of the demonstration in this paper of the interpretation of concepts in mathematics and related disciplines in terms of ICMUP and related concepts (Sections 6 and 7), there is potential for the augmentation and adaptation of mathematics with concepts and mechanisms from the SP System, especially SP-multiple-alignment and unsupervised learning via the building of SP-grammars. Those concepts, with associated ideas, may provide the basis of a *New Mathematics*.

The next subsection summarises some potential benefits of an NM, and the one after that expands on one possibility in how an NM may facilitate the development of scientific theories.

### 9.2.1 Potential benefits of a New Mathematics

Some of the potentially useful things that an NM may do are summarised here:

- *Extending the range of applications for mathematics.* In [66, Section III], it is is argued that the SP System has potential to be developed into a *universal framework for the representation and processing of diverse kinds of knowledge* (UFK). Those arguments apply *a fortiori* to an NM, drawing as it would on the resources of both the SP System and mathematics.

- *Facilitating the integration of scientific theories.* By providing a UFK for the description and processing of related but incompatible theories such as quantum mechanics and relativity, an NM has the potential to help iron out inconsistencies between such theories and to facilitate their integration.

- *Several kinds of reasoning.* Since the SP System already demonstrates strengths and potential in several different kinds of reasoning ([63, Section 10], [61, Chapter 7]), an NM may provide a means of drawing inferences from observations and concepts in science and elsewhere, across a much wider area than is usual now. Potential benefits here include a softening of the boundary between 'exact', all-or-nothing styles of reasoning, and probabilistic kinds of reasoning.

- *Development of scientific theories.* By exploiting strengths of the SP System in unsupervised learning (which, in the future, are likely to be much more fully developed than they are now), an NM may facilitate the automatic or semi-automatic development of scientific theories from data [67, Section 6.10.7]. There is more in Section 9.2.2, below.

- *Quantitative evaluation of scientific theories.* An NM may provide a means of quantifying the Simplicity of any scientific theory, and its descriptive or explanatory Power, and thus, via the ratio of those two measures, facilitating quantitative comparisons amongst rival scientific theories.

- *The study of complex systems.* There seems to be potential for the NM in the study of *complex systems*, meaning systems which "... are characterized by interactions between their components that produce new information—present in neither the initial nor boundary conditions—which limit their predictability."[8]

- *The integration of mathematics, logic, and computing.* In an NM, the distinction between mathematics, logic, and computing would largely disappear in, for example, the SP Machine (Section 3.8).

- *A new perspective on statistics.* Because of the intimate relation between IC and concepts of probability (Section 8), and the apparent advantages in approaching probability via ICMUP (Section 8.2), an NM has the potential to provide a whole new perspective on statistics, with potential advantages over established ideas.

- *Facilitating the learning and use of mathematics.* There seems to be potential to make mathematics more transparent in the representation of fundamentals such as ICMUP and its workings. There is a corresponding potential for mathematics to be easier to learn, to understand, and to use.

---

[8]In "About this Journal" of the journal *Complexity*, bit.ly/2DZ1t3A, retrieved 2019-08-16.

- *New approaches to concepts of 'proof', 'theorem', and related ideas.* There appears to be potential for improved concepts of 'proof', 'theorem', and related ideas, incorporating measures of IC. And there appears to be potential for the automatic or semi-automatic discovery of new theorems and other results in mathematics, and for their automatic or semi-automatic application.

- *Quantification of confidence in inferences.* By contrast with mathematics, where inferences can be, and often are, made without any measure of the confidence that may attach to those inferences, the SP System provides measures of confidence for all its inferences. Thus, for example, inferences of singularities far into the past may be made with much less confidence with the SP System than they are sometimes made with mathematics.

- *Mathematical study of the SP System.* Some parts of the SP System, the concept of SP-multiple-alignment in particular, may prove to be of interest from a mathematical perspective.

### 9.2.2   An additional potential benefit of an NM in science

Apart from the potential benefits to scientific research that are implicit in the more general potential benefits of an NM summarised in Section 9.2.1, there is another kind of potential benefit for science, outlined here.

It appears that, on occasion, leading scientists have developed concepts which they have found difficult to express with the mathematics that was available to them. Here are three putative examples:

- It appears that Michael Faraday developed his ideas about electricity and magnetism with little or no knowledge of mathematics and that James Clerk Maxwell translated them into mathematical form:

    "Without knowing mathematics, [Faraday] writes one of the best books of physics ever written, virtually devoid of equations. He sees physics with his mind's eye, and with his mind's eye creates worlds." [48, Location 623].

  and

    "Maxwell quickly realizes that gold has been struck with [Faraday's] idea. He translates Faraday's insight, which Faraday explains only in words, into a page of equations. These are now known as Maxwell's equations. They describe the behaviour of

the electric and the magnetic fields: the mathematical version of the 'Faraday lines'." [48, Location 677].

- Charles Darwin described his theory of evolution by natural selection with words and pictures. To this day, it is still normally described in that way (but see Gregory Chaitin's proposals for creating "a general, abstract mathematical theory of evolution that captures the essence of Darwin's theory and develops it mathematically." [16, Location 189]).

- It seems that Albert Einstein's ideas were generally developed first in non-mathematical form and only later cast into mathematics:

  "Einstein had a unique capacity to imagine how the world might be constructed, to 'see' it in his mind. The equations, for him, came afterwards; ... For Einstein, the theory of general relativity is not a collection of equations: it is a mental image of the world arduously translated into equations." [48, Location 1025].

Judging by the quotes above, much of the thinking of at least some leading scientists is visual, and not expressible directly in terms of equations (although mainstream mathematics includes visual structures such as 2D and 3D charts, geometrical figures, and topological structures). However, an NM may, in addition, provide a means of representing two-dimensional structures via 2D SP-patterns, and three-dimensional structures via SP-patterns as described in [64, Sections 6.1 and 6.2].

The provision of cognitive structures like those may help scientists to think and communicate directly with NM concepts, without the need to translate their ideas into some less congenial form. It seems possible that an NM may provide the means of representing and processing scientific concepts in forms that are more in accord with the intuitions of scientists like Michael Faraday, Charles Darwin, and Albert Einstein than is conventional mathematics.

There is relevant discussion in José Luis Bermúdez's book on *Thinking Without Words* [8] and Hans Furth's book on *Thinking Without Language: Psychological Implications of Deafness* [21].

## 9.3   The 'Big Picture'

In view of the wide scope of IC, ICMUP, and SP-multiple-alignment (described in Section 3.7, in [63, 61], and in [71]), there is potential for fruitful research exploring their applicability in at least seven areas where the inter-related principles of IC, ICMUP, and SP-multiple-alignment may, with advantage, be applied. This *Big Picture* currently comprises:

- *IC as a unifying principle in AI.* The strengths and potential of the SP System in aspects of human intelligence (Section 3.7) and the central role in the SP System of SP-multiple-alignments (Section 3.3), is evidence for ICMUP as a unifying principle in AI.

- *IC as a unifying principle in HLPC.* A companion to the present paper describes relatively direct empirical evidence for ICMUP as a unifying principle in HLPC [71, Sections 4 to 20]. The strengths and potential of the SP System in modelling aspects of HLPC (Section 3.7), and the central role of SP-multiple-alignment in the SP System, provides further evidence for importance of ICMUP in HLPC [71, Section 21].

- *IC in neuroscience.* Because of its central role in *SP-Neural* (Section 3.6), ICMUP has clear potential in neuroscience, as described in [68].

- *IC, concepts of probability, and statistics.* It is known that there is an intimate relation between IC and concepts of probability (Section 8). But there is an asymmetry between ICMUP and concepts of probability, and there are apparent advantages in approaching probability via ICMUP, as outlined in Section 8.2.

  For those kinds of reasons, there is potential for useful new thinking of methods for statistical analysis in experimental science, epidemiology, social science, and so on.

- *Causation.* Although, in the SP programme of research, little has yet been done on the concept of the causation, it seems likely that useful things can be said about causation in terms of the SP concepts(Section 8.2.4.

- *IC as a foundation for mathematics, logic, and computing.* This paper argues that much of mathematics, perhaps all of it, and much of logic and computing, may be understood as ICMUP. There is potential for the creation of a New Mathematics via the adoption of the SP System as a part of mathematics (Section 9.2). And there are many potential benefits of such a New Mathematics, as outlined in Section 9.2.1.

- *IC as a unifying principle in science.* The importance of Ockham's razor and related ideas is widely recognised by respected scientists. Thus: Isaac Newton writes that "Nature is pleased with simplicity" [37, p. 320]; Ernst Mach [3] and Karl Pearson [44] suggest independently of each other that scientific laws promote "economy of thought"; Albert Einstein writes that "A theory is more impressive the greater the simplicity of its premises, the more

different things it relates, and the more expanded its area of application.";[9] John Barrow writes that: "Science is, at root, just the search for compression in the world" [7, p. 247]; and Ming Li and Paul Vitányi write that: "Science may be regarded as the art of data compression" [32, Section 8.9.2].

As noted in Section 9.2.1, there is potential with the SP System for the the automatic or semi-automatic development of scientific theories from data, and for the quantitative evaluation of scientific theories. The SP System may also serve as a UFK (*ibid.*), perhaps facilitating the encoding of scientific data in a form that accommodates the way in which some leading scientists think (Section 9.2.2).

As with "Strengthening evidence for the SP System" at the beginning of Section 9, the observation that the explanatory range of IC (and ICMUP and SP-multiple-alignment) may, potentially, be extended from AI to five other areas, means that there is potential for an improvement in the ratio of the Simplicity of IC, ICMUP, and SP-multiple-alignment, to their descriptive and explanatory Power. Hence, there is potential for the credibility of these IC-related concepts, in terms of Ockham's razor, to be substantially increased.

Another way of looking at this is to say that the six components of the Big Picture are mutually supportive in the sense that the credibility of any one of them, including the main thesis of this paper, is strengthened via empirical and analytical evidence in support of any and all of the other five of its components.

More specifically, insights or developments in one component may prove useful in one or more of the other components. Here are some examples:

- The ICMUP perspective, SP-multiple-alignment, and unsupervised learning, which are, together, central in the SP System, may prove useful in varied areas of science, perhaps as part of the proposed New Mathematics, with potential benefits outlined in Section 9.2.1.

- The "varied areas of science" just mentioned would of course include neuroscience, another of the six components of the big picture. In that connection, SP-Neural, the 'neural' version of the SP Theory [68], has potential as a source of hypotheses because it draws on the several insights into HLPC, developed in the non-neural version of the SP Theory.

- Further study of the applicability or otherwise of ICMUP concepts (perhaps including SP-multiple-alignment) would be welcome in diverse areas of mathematics.

---

[9]Quoted in [29, p. 512].

# 10 Conclusion

This paper describes a novel perspective on the foundations of mathematics: how much of mathematics, perhaps all of it, may be seen as ICMUP.

ICMUP is itself a novel approach to IC, couched in terms of non-mathematical primitives, as is necessary in any investigation of the foundations of mathematics.

This new perspective on the foundations of mathematics has grown out of: 1) an accumulation of evidence for the importance of IC in HLPC; and 2) a long-term programme of research developing the SP System, meaning the SP Theory of Intelligence and its realisation in the SP Computer Model.

Seven variants of ICMUP are described in Section 5.

In arguing for MICMUP, Section 6 shows first how mathematics may achieve compression of information. Then it shows how variants of ICMUP may be seen in widely-used structures and operations in mathematics.

Section 7, argues, in a similar way, that much of the mathematics-related disciplines of logic and computing may be seen to be founded on ICMUP.

Section 8 describes first how the intimate relation between IC and concepts of probability makes sense in terms of ICMUP, how it relates to the established view that at least one part of mathematics is probabilistic, and how that latter view may be reconciled with the all-or-nothing, 'exact', forms of calculation or inference that are familiar in mathematics, logic, and computing.

Although IC and concepts of probability are closely-related, there is an asymmetry between them: there are advantages in approaching probability via ICMUP and not the other way round (Section 8.2).

Section 9 outlines some of the potential benefits and applications of the ICMUP perspective, in strengthening support for the SP System, as the basis for a proposed New Mathematics (Section 9.2), and as of a 'Big Picture' across AI, HLPC, neuroscience, concepts of probability, mathematics and related disciplines, and as a unifying principle in science.

There are many potential benefits and applications arising from these new perspectives.

# Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

# Acknowledgements

# A    Apparent contradictions of the idea that IC is fundamental in mathematics, AI, and related disciplines

There are two apparent contradictions of the idea that mathematics, AI, and related disciplines may be understood in terms of IC. Those two apparent contradictions and how they may be resolved are described briefly in the following two subsections.

## A.1    The apparent paradox of 'decompression by compression'

The idea that mathematics and related disciplines are largely, perhaps entirely, about compression of information seems to conflict with the undoubted fact that, with some simple mathematics or a simple computer program, it is possible to create data containing large amounts of repetition or redundancy.

This apparent inconsistency may be resolved via the concept of *decompression by compression*, described in [71, Appendix C.1]. In brief, any relatively large 'chunk' of information, such as the expression "Treaty on the Functioning of the European Union" may, in some kind of dictionary, be assigned a relatively short name or 'code' such as "TFEU". Then, in accordance with the chunking-with-codes technique for IC (Section 5.2), compression of the original document may be achieved by replacing each instance of "Treaty on the Functioning of the European Union" with the relatively short code "TFEU".

After that, the original document may be recreated by retrieving each instance of "Treaty on the Functioning of the European Union" via its short code, "TFEU". In that process of retrieval, each instance of "TFEU" in the compressed document must be matched and unified with a copy of that code in the dictionary. Each such case of matching and unification may be seen as a process of compressing information. Hence, decompression of the encoded document has been achieved

via the unification of codes within the document with codes in the dictionary, with a corresponding compression of the information in the codes.

There is a rather loose analogy with the way that, in an old-fashion refrigerator that is powered by gas, it is the heat of a gas flame that, paradoxically, achieves the cooling of the refrigerator.

## A.2 Redundancy is often useful in the storage and processing of information

There is no doubt that informational redundancy—repetition of information—is often useful. For example, it is standard practice in computing to maintain two or more copies of important software and data as a safeguard against the corruption or loss of those things.

With any website that is likely to be viewed by many people, it is normal practice to maintain multiple copies of the website, each with one or more servers, to avoid bottlenecks when many people try to access the website at the same time.

These kinds of uses of redundancy may seem to conflict with the idea that IC— which means reducing redundancy—is fundamental in mathematics and related disciplines.

This issue and how it may be resolved is discussed in [71, Appendix C.2]. In brief, a given body of information $\mathbf{X}$ may be a highly compressed version of another body of information $\mathbf{Y}$. But to ensure that that valuable nugget of information, $\mathbf{X}$, is safeguarded against loss or corruption, any prudent manager of an information system would ensure that there were multiple copies of $\mathbf{X}$, preferably in two or more different locations. Thus he or she would create the apparent contradiction that the multiple copies of $\mathbf{X}$ would be compressed and redundant at the same time.

## B    Why should we assume that the future will be like the past?

With regard to inductive reasoning and its justification mentioned Section 8:

> "We can, of course, ... ask, as philosophers have done for many years: 'What is the rational basis for inductive reasoning?' Why do most people have this strong intuition that because the sun has always risen every morning it will do it again tomorrow, or because every paving stone in a path has held our weight so far, the next one will too? None of these conclusions can be proved logically.

"It is no good arguing that inductive reasoning is rational because it has always worked in the past. This argument eats its own tail. Here is an argument why inductive reasoning is rational which does not depend on the principle which it is trying to justify:

"If we assume that the world, in the future, will contain redundancy in the form of recurring patterns of events, then brains and computers which store information and make inductive inferences will be useful in enabling us to anticipate events. If it turns out that the world, in the future, does indeed contain redundancy, then our investment in the means of storing and processing information will pay off. If it turns out that the world, in the future, does not contain redundancy then we are dead anyway—reduced to a pulp of total chaos!

"This kind of reasoning made fortunes for speculators after World War II: it was rational to buy up London bomb sites during the war because, if the war were won, they would become valuable. If the war were to be lost, the money saved by not making the investment would, in an uncomfortable and uncertain future, probably not be much use anyway." [60, pp. 28–29].

# References

[1] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61:183–193, 1954.

[2] Mueller B and J. Rafelski. *The Structured Vacuum—Thinking about Nothing.* Verlag Harri Deutsch, Frankfurt am Main, 1985.

[3] E. C. Banks. The philosophical roots of Ernst Mach's economy of thought. *Synthese*, 139:25–53, 2004.

[4] D. Barker-Plummer. Turing machines. In E. N. Zalta, editor, *Stanford Encyclopedia of Philosophy.* Winter 2012 edition, 2012. First published 1995-09-14; substantive revision 2012-06-26.

[5] H. B. Barlow. Sensory mechanisms, the reduction of redundancy, and intelligence. In HMSO, editor, *The Mechanisation of Thought Processes*, pages 535–559. Her Majesty's Stationery Office, London, 1959.

[6] H. B. Barlow. Trigger features, adaptation and economy of impulses. In K. N. Leibovic, editor, *Information Processes in the Nervous System*, pages 209–230. Springer, New York, 1969.

[7] J. D. Barrow. *Pi in the Sky*. Penguin Books, Harmondsworth, 1992.

[8] J. L. Bermúdez. *Thinking Without Words*. Oxford University Press, Oxford, 2003.

[9] G. M. Birtwistle, O-J Dahl, B. Myhrhaug, and K. Nygaard. *Simula Begin*. Studentlitteratur, Lund, 1973.

[10] M. Burgin. *Theory of Named Sets*. Nova Science Publishers, New York, 2011.

[11] M. Burgin. *Structural Reality*. Gazelle, Lancaster, uk edition, 2012.

[12] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the Association for Computing Machinery*, 22(3):329–340, 1975.

[13] G. J. Chaitin. Algorithmic information theory. *IBM Journal of Research and Development*, 21(4):350–359, 1977.

[14] G. J. Chaitin. Randomness in arithmetic. *Scientific American*, 259(1):80–85, 1988.

[15] G. J. Chaitin. *Algorithmic Information Theory*. Cambridge University Press, Cambridge, 1997.

[16] G. J. Chaitin. *Proving Darwin: Making Biology Mathematical*. Pantheon Books, New York, Kindle edition, 2012.

[17] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken NJ, Second, Kindle edition, 2006.

[18] K. Devlin. *Mathematics: The Science of Patterns*. Scientific American Library, New York, 1997.

[19] K. Devlin. *Logic and Information*. Cambridge University Press, Cambridge, 2003.

[20] M. Ford. *Architects of Intelligence: the Truth About AI From the People Building It*. Packt Publishing, Birmingham, UK, Kindle edition, 2018.

[21] H. G. Furth. *Thinking Without Language: Psychological Implications of Deafness*. The Free Press, 1966.

56

[22] N. Gauvrit, H. Zenil, and Tegn'er J. The information-theoretic and algorithmic approach to human, animal and artificial cognition. In G. Dodig-Crnkovic and R. Giovagnoli, editors, *Representation and Reality: Humans, Animals and Machines*, pages 117–139. Springer, Heidelberg, 2017.

[23] R. Hersh. *What is Mathematics Really?* Vintage, London, 1997.

[24] D. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid.* Penguin Books, Harmondsworth, 1980.

[25] L. Horsten. Philosophy of mathematics. In *Stanford Encyclopedia of Philosophy.* Stanford University, Stanford, California 94305, 2017.

[26] M. Hutter. Algorithmic information theory. *Scholarpedia*, 2(3):2519. revision #186543.

[27] M. Hutter, S. Legg, and P. M. B. Vitányi. Algorithmic probability. *Scholarpedia*, 2(8):2572. revision #151509.

[28] A. D. Irvine and H. Deutsch. Russell's paradox. In *Stanford Encyclopedia of Philosophy.* Stanford University, Stanford, California 94305, 2016.

[29] W. Isaacson. *Einstein: His Life and Universe.* Pocket Books, London, Kindle edition, 2007.

[30] A. N. Kolmogorov. Three approaches to the definition of the quantity of information. *Problems of Information Transmission*, 1:3–11, 1965.

[31] A. N. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, IT-14(5):662–664, 1968.

[32] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer, New York, 4th edition, 2019.

[33] Ø. Linnebo. *Philosophy of Mathematics.* Princeton University Press, Princeton and Oxford, Kindle edition, 2017.

[34] R. Marcus and M. McEvoy, editors. *An Historical Introduction to the Philosophy of Mathematics.* Bloomsbury Academic, London, Kindle edition, 2016.

[35] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry. ISBN 0-262-63022-2.* MIT Press, Cambridge, Mass., 1969.

[36] E. Nagel and J. R. Newman. *Gödel's Proof.* New York University Press, New York, Kindle edition, 2001.

[37] I. Newton. *The Mathematical Principles of Natural Philosophy*. The Perfect Library, Kindle edition, 2014. First published 1687. Illustrated and bundled with *Life of Sir Isaac Newton*.

[38] N. Nisan and S. Schocken. *The Elements of Computing Systems: Building a Modern Computer from First Principles*. The MIT Press, Cambridge, Mass., Kindle edition, 2005.

[39] V. Palade and J. G. Wolff. A roadmap for the development of the 'SP Machine' for artificial intelligence. *The Computer Journal*, 62, 2019. arXiv:1707.00614, bit.ly/2tWb88M.

[40] C. Parsons. Mathematical intuition. In *Proceedings of the Aristotelian Society*, volume 80, pages 145–168, 1979.

[41] C. Parsons. *Philosophy of Mathematics in the Twentieth Century*. Harvard University Press, Cambridge, Mass., Kindle edition, 2014.

[42] A. Paseau. Naturalism in the philosophy of mathematics. In *Stanford Encyclopedia of Philosophy*. Stanford University, Stanford, California 94305, 2013.

[43] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge, 2000.

[44] K. Pearson. *The Grammar of Science*. Walter Scott, London, 1892. Republished by Dover Publications, 2004, ISBN 0-486-49581-7. Internet archive: bit.ly/1g2gNfk.

[45] R. Penrose. *The Emperor's New Mind*. Oxford University Press, Oxford, 1989.

[46] E. L. Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65:197–268, 1943.

[47] M. D. Resnik. *Mathematics As a Science of Patterns*. Clarendon Press, Oxford, Kindle edition, 1997.

[48] C. Rovelli. *Reality Is Not What It Seems: The Journey to Quantum Gravity*. Penguin Books, London, Kindle edition, 2016.

[49] B. Russell. Letter to frege. In Jean van Heijenoort, editor, *From Frege to Gödel*, pages 124–125. Harvard University Press, Cambridge, Mass., 1967 edition, 1902.

[50] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, Amsterdam, 2012.

[51] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, 1949.

[52] S. Shapiro. *Philosophy of Mathematics*. Oxford University Press, New York, 1997.

[53] R. J. Solomonoff. A formal theory of inductive inference. Parts I and II. *Information and Control*, 7:1–22 and 224–254, 1964.

[54] R. J. Solomonoff. The discovery of algorithmic probability. *Journal of Computer and System Sciences*, 55(1):73–88, 1997.

[55] W. J. Teahan and K. M. Alhawiti. Preprocessing for PPM: compressing UTF-8 encoded natural language text. *International Journal of Computer Science & Information Technology*, 7(2):41–51, 2015.

[56] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[57] M. van Atten. *On Brouwer*. Wadsworth, London, 2004.

[58] C. S. Webster. Alan turing's unorganized machines and artificial neural networks: his remarkable early work and future possibilities. *Evolutionary Intelligence*, 5:35–43, 2012.

[59] E. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics*, 13:1–14, 1960.

[60] J. G. Wolff. *Towards a Theory of Cognition and Computing*. Ellis Horwood, Chichester, 1991.

[61] J. G. Wolff. *Unifying Computing and Cognition: the SP Theory and Its Applications*. CognitionResearch.org, Menai Bridge, 2006. ISBNs: 0-9550726-0-3 (ebook edition), 0-9550726-1-1 (print edition). Distributors, including Amazon.com, are detailed on bit.ly/WmB1rs.

[62] J. G. Wolff. Towards an intelligent database system founded on the SP theory of computing and cognition. *Data & Knowledge Engineering*, 60:596–624, 2007. arXiv:cs/0311031 [cs.DB], bit.ly/1CUldR6.

[63] J. G. Wolff. The SP Theory of Intelligence: an overview. *Information*, 4(3):283–341, 2013. arXiv:1306.3888 [cs.AI], bit.ly/1NOMJ6l.

[64] J. G. Wolff. Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision. *SpringerPlus*, 3(1):552–570, 2014. arXiv:1303.2071 [cs.CV], bit.ly/2oIpZB6.

[65] J. G. Wolff. Autonomous robots and the SP Theory of Intelligence. *IEEE Access*, 2:1629–1651, 2014. arXiv:1409.8027 [cs.AI], bit.ly/18DxU5K.

[66] J. G. Wolff. Big data and the SP Theory of Intelligence. *IEEE Access*, 2:301–315, 2014. arXiv:1306.3890 [cs.DB], bit.ly/2qfSR3G. This paper, with minor revisions, is reproduced in Fei Hu (Ed.), *Big Data: Storage, Sharing, and Security*, Taylor & Francis LLC, CRC Press, 2016, Chapter 6, pp. 143–170.

[67] J. G. Wolff. The SP Theory of Intelligence: benefits and applications. *Information*, 5(1):1–27, 2014. arXiv:1307.0845 [cs.AI], bit.ly/1FRYwew.

[68] J. G. Wolff. Information compression, multiple alignment, and the representation and processing of knowledge in the brain. *Frontiers in Psychology*, 7:1584, 2016. arXiv:1604.05535 [cs.AI], bit.ly/2esmYyt.

[69] J. G. Wolff. The SP Theory of Intelligence: its distinctive features and advantages. *IEEE Access*, 4:216–246, 2016. arXiv:1508.04087 [cs.AI], bit.ly/2qgq5QF.

[70] J. G. Wolff. Software engineering and the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2017. Submitted for publication. arXiv:1708.06665 [cs.SE], bit.ly/2w99Wzq.

[71] J. G. Wolff. Information compression as a unifying principle in human learning, perception, and cognition. *Complexity*, 2019:38 pages, February 2019. Article ID 1879746. viXra:1707.0161v3, hal-01624595 v2.

[72] J. G. Wolff. Unsolved problems in ai, described in the book *architects of intelligence* by martin ford, and how they may be solved via the sp system. 2019. Submitted for publication. bit.ly/2th7Bze.

[73] H. Zenil and N. Gauvrit. Algorithmic cognition and the computational nature of the mind. In R. A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 1–9. Springer, New York.