How a *New Mathematics* may with advantage be applied in science

J Gerard Wolff*

July 2, 2021

Abstract

Keywords: information compression; SPMA; mathematics; quantum mechanics; artificial intelligence.

1 Introduction

This paper describes a proposed *New Mathematics* NM and how it may with advantage be applied in science.

Although the same name has been used for other initiatives,[1] this concept of NM was first proposed in [41, Section 9]. This paper is intended as an expanded and revised version of the VM concept with an exploration of its potential benefits and applications.

In brief, the VM is intended as an amalgamation of mathematics with the *SP System* (SPS), where the SPS is the *SP Theory of Intelligence* and its realisation in the *SP Computer Model* (SPCM). The proposed amalgamation is described in Section 3.1, below.

Development of the NM may seem like a monstrous upsetting of the mathematical applecart that has worked well for hundreds of years. But there are several potential benefits from the NM, most of which would be

---

*Dr Gerry Wolff BA (Cantab) PhD (Wales) CEng MIEEE MBCS; ORCID ID: 0000-0002-4624-8904; CognitionResearch.org, UK; jgw@cognitionresearch.org; +44 (0) 1248 712962; +44 (0) 7746 290775; *Twitter*: @gerrywolff65; *Web*: www.cognitionresearch.org.

[1] Such as a method for teaching mathematics described in *New Math* in Wikipedia, tinyurl.com/ma2fs5c8, retrieved 2021-06-18.

helpful in scientific research. Of course the NM may be useful in non-scientific applications but those would be matters to be discussed elsewhere.

Abbreviations used in this paper are defined in Appendix E. In that appendix, the origins and proposed use of the name 'SP' are also described.

The sections of the paper are described briefly here. Five introductory sections come first:

- *Origin of the proposals* (Section 2.1). This section summarises how these proposals originated in research in AI and human learning, perception, and cognition (HLPC).
- *Seven techniques for information compression, and their potential* (Section 2.2). This section describes seven techniques for information compression (IC) which provide the foundation for the proposals.
- *Information compression in mathematics* (Section 2.3). This outlines the arguments that much of mathematics, perhaps all of it, may be understood as a set of techniques for information compression (IC), and their application.
- *Information compression and concepts of inference and probability* Section 2.5. There is an intimate relation between IC and concepts of inference and probability.
- *Towards the development of a New Mathematics* (Section 3.1). Here, the main elements of the proposed NM are described.

These are followed by several main sections describing potential benefits of the NM:

- *Providing support for thinking about theoretical issues by scientists and others* (Section 5.10). How the NM may provide support for 'non-mathematical' kinds of thinking employed by scientists and others.
- *Adding an AI dimension to mathematics* (Section 4). The NM may provide an AI dimension to mathematics with many potential benefits.
- *Facilitating the simplification and integration of mathematics, logic, and computing* (Section **??**). There is clear potential for the simplification and integration of mathematics, logic, and computing.
- *Development of the New Mathematics as a Universal Framework for the Representation and Processing of Knowledge* (Section 3.2).

- *A new perspective on statistics?* (Section 6). There appears to be potential for a new perspective on statistics.
- *Infinity in mathematics and science* (Section **??**). The NM may help to tame problems with infinities that are recognised in physics and cosmology.
- *Superposition* (Section 8). It seems that the NM may have some useful things to say about the concept of superposition in quantum mechanics (QM).
- *Nonlocality, entanglement, SPMA, and discontinuous dependencies* (Section 9). It seems that the NM may have some useful things to say about the concepts of nonlocality and entanglement in quantum mechanics.

2 Preliminaries

2.1 Origin of the proposals

A guiding principle in the creation of the VM is the importance of information compression (IC) in human learning, perception, and cognition (HLPC). This strand of research was pioneered by Fred Attneave (see, for example, [2, 3]) and Horace Barlow (see, for example, [5, 6]) and is still continuing. There is a review of relevant evidence in [40].

IC has also been a guiding principle in the development of the SPS, since the SPS is, amongst other things, an attempt to model HLPC. There is an outline of the SPS in Appendix **??** with pointers to where more information may be found.

Since mathematics is designed to help human thinking, and is the product of human brains, it should not be surprising to find similar principles in mathematics. In another paper [41], it is argued that *much of mathematics, perhaps all of it, may be seen as a set of techniques for IC, and their application.*

This has given rise to the idea, outlined in [41, Section 9], that there are potential advantages in creating a *New Mathematics* as an amalgamation of mathematics and the SPS, with IC at centre stage.

2.2 Seven techniques for IC

In this research, seven techniques for IC are recognised, with potential for more. This subsection, which draws on [41, Section 5], summarises thinking in this part of the SP Theory of Intelligence.

2.2.1 Why not adopt established ideas about information compression?

Before getting on to the seven techniques for IC, a few words are needed about why, in this research, established techniques for IC are not being used.

ICMUP as an alternative to mathematically-oriented techniques for IC. The basis for the seven techniques is the relatively primitive concept of 'IC via the Matching and Unification of Patterns' (ICMUP), as described in Section 2.2.3, below. This approach has been chosen in preference to mathematically-oriented techniques such as arithmetic compression (see, for example, [18, Chapter 6]) or transform methods including wavelet compression (see, for example, [18, Chapter 10]), for three main reasons:

- The organisation and workings of the SPS was inspired in part by the concept of 'multiple sequence alignment' in bioinformatics, and that suggested that IC in the SPS may be achieved entirely via ICMUP.
- Partly for that reason and partly because of the arguments that mathematics, logic, and computing, may be understood in terms of ICMUP [41], a working hypothesis in this research is that all kinds of redundancy may be understood in terms of ICMUP and discovered by methods based on ICMUP.
- If one is to argue that mathematics may be understood in terms of IC (as is done in [41]) it is not appropriate to use models of IC that depend on mathematics. Something more primitive is needed, and ICMUP seems to meet that need.[2]

ICMUP as an alternative to MLE concepts. For similar reasons, concepts of minimum-length encoding (MLE, see, for example, [22]) have not been adopted although, in the spirit of MLE, it is certainly useful on occasion to think of the information content of a body of information, **I**, as the length of the shortest program that one may create that will produce **I**. Also, Solomonoff's development of Algorithmic Probability Theory [27, 28], which is part of the foundations of MLE research, is highly relevant to the NM and its development (Appendix D.2).

---

[2] That said, some mathematics is used in the SPCM and has contributed to its development, as described in Appendix D. But those uses of mathematics, which are at fairly low levels in the SPCM, does not change the ICMUP-based techniques which are central in its workings.

paragraphICMUP as an alternative to the concept of universal Turing machine. Another reason for not basing the SPS more directly on MLE concepts is that those concepts are based on the concept of a universal Turing machine as a definition of computing whereas the SP Theory is itself a definition of computing [38, Section II-C].

2.2.2 The importance of heuristic search

In all but the smallest bodies of information, **I**, there is normally an astronomically large number of ways in patterns may be matched and unified. Accordingly: it is not feasible to search exhaustively for the matches and unifications that yield the greatest value for IC; it is necessary to use heuristic methods, trading accuracy for speed; and it is not possible to prove that the best possible result has been obtained.

2.2.3 Basic ICMUP

The first of the techniques for IC to be described is the idea, already mentioned, of finding patterns that match each other and the merging or 'unification' of two or more matching patterns to make one. This is illustrated in the top part of Figure 1 which shows how two instances of the pattern '`INFORMATION`' may be unified to make a single instance.

In this connection, the expression "IC via the Matching and Unification of Patterns" may be abbreviated as 'ICMUP'.

The example in Figure 1 may give the impression that the concept of 'pattern' in this research means a coherent array of symbols like '`INFORMATION`'. But in the SPS, the concept of 'pattern' includes arrays of symbols in which other symbols may be interspersed, such as '`INpFORpqrMATstION`'. The development of the SPS has been correspondingly more challenging, but also more rewarding.

As we shall see, the ICMUP concept is bedrock in the other six techniques for IC described in this section.

A working hypothesis in this research is that ICMUP is fundamental in *all* kinds of IC, including mainstream techniques such as arithmetic coding and wavelet encoding (see Appendix 2.2.1).

2.2.4 Chunking-with-codes

A problem with Basic ICMUP is that, if two or more patterns within a body of information, **I**, are unified, information is lost about the locations of those patterns within **I**, assuming that the unified pattern is stored in a separate dictionary of patterns.

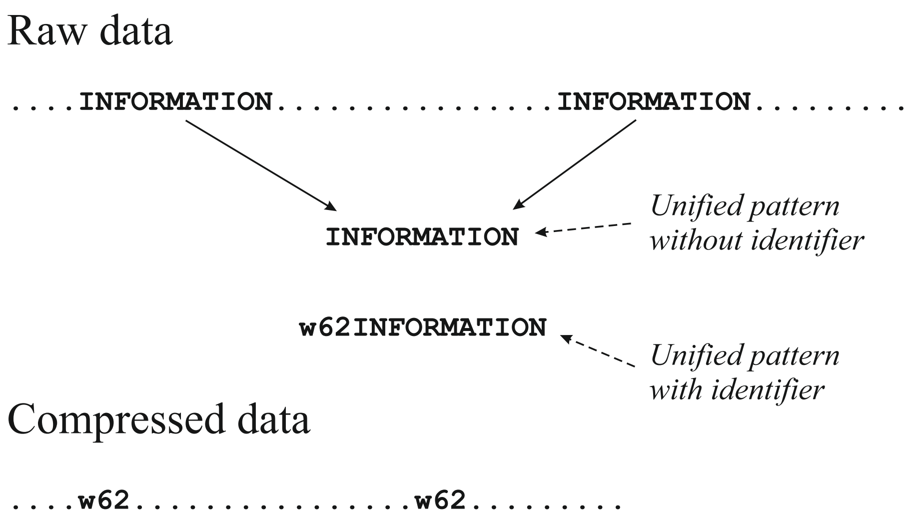

Figure 1: A schematic representation of the way two instances of the pattern '`INFORMATION`' in a body of data may be unified to form a single 'unified pattern', shown just above the middle of the figure. To achieve lossless compression, the relatively short identifier '`w62`' may be assigned to the 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 [30], with permission.

A long-established answer to this problem, called 'chunking-with-codes', is to assign a relatively brief identifier or 'code' to the unified pattern (which may be called a 'chunk' of information), and put a copy of that code at each place within **I** that a copy of the chunk occurred before unification. This is illustrated in the lower part of Figure 1 where the identifier '`w62`' is assigned to the unified pattern '`INFORMATION`', and a copy of that identifier is put in each of the original locations of '`INFORMATION`'.

2.2.5 Schema-plus-correction

The schema-plus-correction technique for IC is like chunking-with-codes except that there may be variations in the chunk from one occasion to another. This is achieved by providing for 'corrections' to the chunk that may vary from one occasion to another.

A popular example is a menu in a restaurant or cafe where the menu is the chunk and corrections to the chunk are the different dishes that may be chosen at different locations with the chunk, such as 'soup' or 'prawn cocktail', etc, for the starter, 'roast beef' or 'vegetarian lasagne', etc, for the main course, and 'apple crumble' or 'treacle tart', etc, for the pudding. Of course, the code for the whole menu in this case is something like 'menu', 'bill of fare', etc.

2.2.6 Run-length coding

Run-length coding can be applied when there is a sequence of two or more patterns that match each other. In that case, it is only necessary to identify the pattern once, and either specify the number of repetitions, or show in some way that the given pattern is repeated until there is a signal for the repetition to stop.

In everyday life, we use run-length coding all the time: we may ask our local newsagent to keep delivering a newspaper until we tell them to stop; the pattern 'Sunday ... Saturday' keeps repeating *ad infinitum*, and likewise 'January ... December'; and so on.

2.2.7 Class-inclusion hierarchies

In describing things in the world, we often make use of class-inclusion hierarchies: animals and plants are divided into classes and subclasses through many levels; there are many classes and subclasses of road vehicle and there is often cross-classification as for example where a bus is classified as a passenger vehicle and it is also classified as a heavy vehicle; people are divided into many classes and, as with road vehicles, there is often cross-classification as, for example, when a person may be both a woman and a doctor; foods come in many classes and sub-classes; and so on.

Where does ICMUP come in? If two or more entities have characteristics that match each other and can be unified, those two or more entities may form a new class. And if an entity has characteristics that match those of an existing class (which it does not already belong to), those matching characteristics may be unified so that the entity becomes a member of the class. The overall effect of these matches and unifications is to compress the information in those matching characteristics.

These matches and unifications mean that lower-level classes in a hierarchy do not, in themselves, show all their attributes. But this does not matter because, in a class-inclusion hierarchy, classes at any level except the top level are seen to *inherit* the attributes of all the classes above them in the hierarchy.

2.2.8 Part-whole hierarchies

As with class-inclusion hierarchies, we often describe things in the world in terms of part-whole hierarchies, often through many levels: an animal may be divided into its head, its body, and its legs; the head may be divided into skull, brain, eyes, ears, and so on.

As with class-inclusion hierarchies, ICMUP may be seen in the way part-whole hierarchies originate or are modified. If, for example, two or more cars have the same high-level structure, it only necessary to describe that high-level structure once, and the same applies to all the lower-level structures such as engines, wheels, etc. Naturally, there may be variations but, at any one level, the variations can be described without it being necessary to describe the whole level.

2.2.9 SP-multiple-alignment

The concept of SP-multiple-alignment (SPMA) is described in Appendix A.5. The SPMA concept has a special status in the SPS because it is largely responsible for the versatility and potential of the SPS in AI-related structures and functions (Appendix A.7), including the way it can facilitate the seamless integration of diverse aspects of AI, in any combination.

There seem to be two main reasons for this versatility:

- The SPMA concept may be seen as a generalisation of the six techniques for IC described in the preceding six subsections, a generalisation which means that, any or all of those six techniques may be expressed within an SPMA, in any combination.

There is detailed evidence for this feature of the SPMA concept in Appendix B.

- The SPMA concept is also significant because it provides a powerful means of expressing patterns of redundancy in data, which may in principle have any desired level of accuracy (Appendix A.5.4).

2.2.10 Other possibilities

What has been described in Section 2.2 does not exhaust the potential for finding matches between patterns, and thus for the detection of new kinds of redundancy. Other possibilities include: treating SP-patterns as if they were unordered sets; searching for redundancies between a set of SP-patterns and mirror images of those SP-patterns; searching for redundancies in two-dimensional SP-patterns; and other possibilities described in Appendix C.

2.3 Information compression in mathematics

This section outlines the main arguments in [41] that much of mathematics, perhaps all of it, may be understood as a set of techniques for IC, and their application:

- *Mathematics as a means of compressing information.* Mathematics, in itself and without any special techniques, provides a very effective means of 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, ... 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." [41, p. 13].

- *Chunking-with-codes*:

  > "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." [41, p. 14].

- *Run-length coding, addition*:

> "Since all numbers with bases above 1 may be seen to be compressed representations of unary numbers [41, 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." [41, pp. 14–15].

- *Run-length coding, multiplication*:

> "Multiplication [may be seen as] 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 '$\times 10$' 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." [41, p. 15].

- *Combinations of IC techniques 1*:

> "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." [41, p. 15].

- *Combinations of IC techniques 2*:

> "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 Δt by $\sqrt{1 - v^2/c^2}$." [41, p. 15].

2.4 Information compression in logic and computing

In [41, Section 7], it is argued that arguments that are similar to those outlined in Section 2.3 may be applied to argue that, like mathematics, logic and computing may be understood as a set of techniques for the compression of information, and their application.

It may be thought that, since computers are often used as vehicles for mathematics and logic, they would be mere shadows of those two disciplines with nothing that is distinctive.

An apparent contradiction of that idea is the concept of 'object-oriented programming' (OOP) or 'object-oriented design' (OOD) which was originated in the Simula programming language [7] and is now a feature of most of the widely-used programming languages today.

Since OOP/OOD is essentially a version of class-inclusion hierarchies, the fifth of the seven techniques for IC described in Section 2.2 and [41, Section 5], one might expect it to be prominent in mathematics and logic. But it is altogether absent in those two disciplines or, at best, rarely used.

2.5 Information compression and concepts of inference and probability

Due mainly to the development of Algorithmic Probability Theory (APT) by Ray Solomonoff [27, 28], it has been recognised for some time that there is a very close relationship between concepts of IC and concepts of inference and probability.

Since mathematics appears to be a set of techniques for IC and their application (Section **??**), and because of the close relation between IC and concepts of probability described in the previous paragraph, the same appears to be true of mathematics. [41, Section 8].

At first sight, this is nonsense because of the 'clockwork' non-probabilistic nature of things like $2 + 2 = 4$. But it appears that, at some 'deep' level, number theory, a fundamental part of mathematics, has been shown to be fundamentally probabilistic. In that connection, 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." [10, p. 80].

As indicated in this quotation, randomness in number theory is closely related to Gödel's incompleteness theorems. These are themselves closely related to the phenomenon of recursion, a feature of many formal systems (including the SP System), many of Escher's pictures, and much of Bach's music, as described in some detail by Douglas Hofstadter in his book *Gödel, Escher, Bach: An Eternal Golden Braid* [19].

If, for these kinds of reasons, mathematics is fundamentally probabilistic, the same will be true of the NM and, because the SPCM supports recursion, it will be true of the SPS.

3 The proposed *New Mathematics* and related ideas

3.1 Towards the development of a *New Mathematics*

As noted in the Introduction (Section 1), the creation of an NM was first proposed in [41, Section 9.2]. There it is suggested that:

> "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." and that "those concepts, with associated ideas, may provide the basis of a *New Mathematics* (NM)." [41, p. 20].

In brief, the proposed NM may be developed like this:

1. In broad terms, the NM may be an amalgamation of mathematics and the SPS, both now and as they may evolve in the future.

   Regarding that last point, it is assumed that the SPS will have been developed to become the *SP Machine* (Appendix A.12) as described in [25], and that in particular there will be robust solutions to key problems noted in [32, Section 3.3]: 1) The concept of SP-pattern will have been generalised to two dimensions, whilst retaining one-dimensional SP-patterns (Appendix A.4); 2) How the system may learn to recognise perceptual features in speech and visual images; 3) Other weaknesses in unsupervised learning will have been remedied; and 4) There will be coherent, principled solutions to how the system may process numbers.

2. While [41, Section 6] shows, with evidence, that much of mathematics, perhaps all of it, may be understood in terms of a set of techniques

for IC, there is evidence in [41, Section 7] that similar principles may be seen in the structures and workings of logic and computing. Taken together, these two bodies of evidence suggest the possibility that mathematics, logic, and computing, may be integrated as a single computer system, with IC as a unifying theme. That single system, which we may call *MLC* (short for 'mathematics and logic and computing') would be a useful first step in creating the NM.

3. As a next step, MLC may be integrate with structures and mechanisms from the SPCM, including structures and mechanisms for IC via the building of SPMAs ([32, Section 4] and [30, Sections 3.4 and 3.5]), and for IC via the creation of SP-grammars ([32, Section 5] and [30, Chapter 9]).

4. To avoid creating an NM in which ordinary mathematics looks too strange and unfamiliar to people who are familiar with ordinary mathematics and ordinary logic, there is probably a case for creating 'syntactic sugar' so that NM structures corresponding to ordinary mathematics and logic may be presented in conventional form, with NM structures hidden from view but accessible as required. However, any such syntactic sugar would not change the fundamentals of the system.

5. The vehicle for the NM's representation and processing of data would be the SP Machine, as outlined in Appendix A.12 and illustrated schematically in Figure 7.

3.2 Towards a universal framework for the representation and processing of knowledge, and the standardisation of knowledge

It has been suggested in [35, Section III-A] and [41, Section 5], that the SPS has potential to be developed into a *Universal Framework for the Representation and Processing of Knowledge* UFK. With the addition of mathematics, logic, and computing (items 2 and 3 in Section 3.1), the NM may also be seen to have potential as a UFK.

There is much about AI in the concept of a UFK but it is distinctive in its focus on knowledge and the problems of storing and processing diverse kinds of knowledge.

Potential benefits of the NM as a UFK are described in the next-but-one subsection and those that follow.

3.3 Towards standardised knowledge

Before we get on to those potential benefits, an important adjunct to the concept of UFK is the concept of *Standardised Knowledge* (STK). The UFK is a framework for representing knowledge, while the STK is specific knowledge.

For example, a good UFK should provide an effective means of representing the syntax and semantics of any natural language, while an STK in that area may specify meanings and internationally-agreed forms in which they should be represented across the multitude of things that people may wish to talk or write about.

An example of that kind of standardisation is the way in which dates and times should be represented, recommended by ISO, the International Organization for Standardization. They suggest, with reasons, that the form should be YYYY-MM-DD hh:mm:ss rather than the many other forms in use.

As described in Section 4.5, this kind of standardisation can be very useful in artificial systems for translating between natural languages.

4 Adding an AI dimension to the NM

This section and those that follow describe features of the NM that may prove useful in science.

Since the NM will incorporate the SPS (as outlined in Appendix A), it will have the strengths and potential of the SPS in aspects of intelligence, as outlined in Appendix A.7, and summarised even more briefly here:

- Intelligence-related capabilities such as unsupervised learning (more in Section 3.2), the analysis and production of natural language, and so on.
- In particular, several forms of reasoning such as one-step 'deductive' reasoning, chains of reasoning, abductive reasoning, and so on.
- The representation and processing of intelligence-related forms of knowledge.
- The seamless integration of diverse aspects of intelligence-related capabilities and forms of knowledge, in any combination.

  As noted in Appendix A.7.4, the seamless integration of diverse kinds of intelligence-related capabilities and intelligence-related knowledge appears to be *essential* in achieving human-level broad AI. In scientific research, that kind of fluidity and adaptability is likely to be at least as useful in any artificial system as it is in flesh-and-blood scientists.

4.1 Helping to overcome the problem of variety in data

A major potential benefit of a UFK would be reducing the hundreds of kinds of data and kinds of application, each with a distinct file extension, as listed in the Wikipedia articles *List of file formats*,[3] and *List of filename extensions*.[4] Although this may seem to be an overambitious claim, there is potential in the SPS and NM to reduce the variety of kinds of data to 1, and to reduce the variety of kinds of application to 1. Here's how:

- *One kind of data.* Anything to be processed as 'data', regardless of what it is about, would be a set of New SP-patterns.
- *One kind of application.* The SPCM or its development as the SP Machine may be regarded as one application, the only one in the SPS. It is true that Old SP-patterns can influence the workings of that one application, but all Old SP-patterns are processed in the same way.

4.2 Facilitating the discovery of structure in data

Unsupervised learning in the SPCM, described in [32, Section 5] and [30, Chapter 9] has demonstrated how it is possible to discover SP-grammars in data, including relatively concrete structures like words and more abstract structures like the abstract form of a sentence. We may reasonably expect that future versions of these learning processes, perhaps embodied in the SP Machine, will be more robust and capable of discovering meaningful structures in a wide variety of kinds of data.

As noted elsewhere there is also potential to discover three-dimensional structures from several partially-overlapping two-dimensional views of an object (Appendix A.4), and also to discover three-dimensional structures by merging pairs of two-dimensional SP-patterns (Appendix C.2).

4.3 The interpretation of data in terms existing knowledge

With appropriate New SP-patterns and Old SP-patterns, the SPCM can parse sentences as described in Appendix A.5. This is a form of interpretation

---

[3] tinyurl.com/2efzsuyc, retrieved 2021-06-26.

[4] tinyurl.com/n8u3sjb9, retrieved 2021-06-26.

of the New information in terms of the Old information but falls short of the analysis of a sentence for its meanings.

Preliminary work in that area, described in [30, Section 5.7], shows how, with a simple sentence, the SPCM can derive a meaning from the surface form of a sentence [30, Figure 5.18], and how the surface form of a sentence may be derived from meanings [30, Figure 5.19]. Clearly, there is more work to be done in this area.

4.4 Data fusion

If two or more bodies of data are to be merged or fused, it clearly helps if there is uniformity in how information such as dates, weights, distances, and so on, are expressed. With a mapping between surface forms and a standardised form for knowledge (STFK), the SPS and the NM have potential to provide the necessary preprocessing.

4.5 As an interlingua to facilitate translation between natural languages

An STFK, as just described but with extensive knowledge of meanings, may serve as an *interlingua* for translations between different languages, so that translations can work with meanings. Ultimately, this is likely to prove more accurate and generally more effective than computer-translation systems that work with the surface forms of different languages.

4.6 To facilitate the standardisation required for the "Semantic Web", and the "Internet of Things"

As with data fusion and translations between languages, an STFK with appropriate knowledge is likely to prove useful in setting up and operating such things as the "Semantic Web",[5] and the "Internet of Things".[6].

4.7 Helping to minimise the problem of obsolescence in data and its formats and formalisms

Again, an STFK would help to reduce problems of obsolescence on formats and formalisms for data. This presupposes that all such formats and formalisms are stored with translations into STFK meanings. Then, if a partic-

---

[5] See *Semantic Web* in Wikipedia, retrieved 2021-06-30.

[6] See *Internet of things* in Wikipedia, retrieved 2021-06-30.

ular format or formalism falls out of use, it should be possible, with STFK mappings, to interpret any older documents that use it.

4.8 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."[7]

5 Potential for improvements in scientific theories

5.1 Creation

5.2 representation

5.3 evaluation

5.4 integration

5.5 Expanding the role of mathematics in the representation and processing of scientific knowledge

The NM may expand the role of mathematics in the representation and processing of all kinds of knowledge, including scientific knowledge. Instead of being confined to the representation of a few brief formulae, the NM may potentially serve in the representation of processing of knowledge that may otherwise be represented with pictures, diagrams, and even verbal descriptions.

In particular, there seems to be potential for the NM to imitate the 'mind's eye' view of scientific knowledge that seems often to be a feature of the thinking of leading scientists (Section 5.10).

---

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

5.6 Unsupervised learning and the development of scientific theories

Although the SPCM can demonstrate unsupervised learning ([30, Chapter 9]), it needs further development as outlined in [32, Section 3.3]. Since science may be seen as a process of gathering information and compressing it, there is potential for the automatic or semi-automatic creation of scientific theories via unsupervised learning from appropriate data [36, Section 6.10.7].

5.7 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.

5.8 Quantitative evaluation of scientific theories

In scientific research as it has been up to now, the evaluation of rival scientific theories has been done via more-or-less informal debate, and it seems likely that this will be true for some time to come. But there is potential for the NM to achieve quantitative evaluation of scientific theories in terms of IC.

5.9 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 probability for all its inferences (Appendix A.5.3).

5.10 Providing support for 'visual' thinking of scientists and others

This section, which is based on [41, Section 9.2.2], suggests that the NM may provide support for the kind of 'visual' or 'mind's eye' thinking of scientists and others, a kind of thinking that can be difficult to express with ordinary mathematics, notwithstanding the existence of 'graphic' branches of mathematics such as geometry or topology.

Here are three possible 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." [**?**, 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'." [**?**, 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." [**?**, 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." [**?**, 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 [**?**, 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* [**?**] and Hans Furth's book on *Thinking Without Language: Psychological Implications of Deafness* [**?**].

6 A new perspective on statistics?

Within mathematics, statistical theory is well established and has proved its worth in many applications in science and elsewhere. But of course there is always room for other perspectives.

As we have seen, there is an intimate relation between IC, probabilities, and mathematics because, if it is accepted that there is an intimate relation between IC and concepts of probability (Section 2.5), and if it is accepted that mathematics may be understood in terms of IC [41], then there should be an intimate relation between mathematics and concepts of probability. Hence we may expect 'normal' mathematics, and the NM, to be fundamentally probabilistic.

In the points that follow, the potential of the NM in statistics is largely because of the potential of the SPS in statistics and the incorporation of the SPS within the NM:

- *Strengths of the SPS with inferences and probabilities.* The strengths of the SPS in the making of inferences and the calculation of associated probabilities (Appendix A.5.3) flow directly from the central role of IC in the workings of the SPS, and because of the intimate relation between IC and concepts of probability (Section 2.5).

- *Several kinds of probabilistic reasoning.* More specifically, the SPMA concept within the SPS has proved to be a powerful vehicle for several kinds of probabilistic reasoning ([32, Section 10], [30, Chapter 7]), and for their seamless integration in any combination (Appendix A.7.4). Collectively, these several kinds of probabilistic reasoning, working together, have potential as a powerful aid to statistical inference.

- *Exploiting the asymmetry between IC and concepts of probability.* Because of the asymmetry between IC and concepts of probability [41, Section 8.2], there are potential advantages in working from IC to probabilities, and not the other way round.

- *Statistical analysis via unsupervised learning.* It appears that, in effect, compression of incoming data via unsupervised learning in the SPS achieves a thorough statistical analysis of those data.
- *Making good use of small frequencies.* As noted in [41, Section 8.2.3], it is possible with ICMUP to exploit situations where frequencies as low as 2 or 3 can be statistically significant. Since ICMUP is bedrock in the building of SPMAs.
- *Modelling Bayesian networks via the SPS.* The SPS has proved to be an effective alternative to Bayesian reasoning, including reasoning in Bayesian networks ([32, Section 10.2], [30, Section 7.8]).
- *Learning structures via probabilistic associations.* In addition to its strengths in learning patterns of association, the SPS, via unsupervised learning, has strengths and potential to learn entire structures, including the potential to learn three-dimensional structures (Appendix C.2 (items 3 and 4) and Appendix A.4). In its strengths and potential for the learning of structures, it goes beyond mainstream statistics.

7 New concepts of "proof," "theorem," and related ideas

As noted in [41, Section 9.2.1], there is potential in the development of the NM for the creation of new concepts of proof, theorem, and related ideas. Because IC is central in the workings of the SPS, it is likely that such developments will incorporate IC, and corresponding measures of probability, as indicators of success. There is also potential for the integration of such concepts with concepts of probabilistic reasoning, as described in [32, Section 10] and [30, Chapter 7].

Concepts like these have many potential applications in science.

8 Superposition

This section presents arguments that the meaning of the word 'superposition' is different from the meaning of that word in wave mechanics, and is essentially the same as the meanings of the expressions ‘syntactic class’ in theoretical linguistics, and ‘data type’ in mathematics and computing.

In what way is the SP relevant? That second concept of ‘superposition’ can be represented and processed by the SPCM in a very straightforward manner, as will be explained.

But first we need to untangle some apparent misconceptions about the concept of superposition and associated ideas in QM.

8.1 Some apparent misconceptions about ‘superposition’ and related ideas in quantum mechanics

The concept of superposition of waves is described by Al-Khalili like this:

> “The idea of superposition is not unique to quantum mechanics but is a general property of all waves. Imagine watching someone dive into an empty swimming pool. You will see the ripples travel outwards along the surface of the water as simple undulations all the way to the other end of the pool. This is in stark contrast to the state of the water when the pool is full of people swimming and splashing about. The turbulent shape of its surface is now due to the combined effect of many disturbances and is achieved by adding them all together. This process of adding different waves together is known as superposition.” [1, location 1025].

This concept of superposition of waves is clear. But *it appears that, in QM, the word ‘superposition’ means something different from what it does in the analysis of waves.* And it appears that this can lead to confusion, both in the concept of ‘superposition’ itself, and in associated concepts such as ‘wavefunction’ and the ‘collapse’ of a wavefunction.

Possible sources of confusion in QM are described in Sections 8.1.1 to 8.2 that follow, drawing on Ball’s book *Beyond Weird* [4].

After that, Sections 8.2.1 and 8.2.2 argue that the concept of superposition is not unique to QM, and much less puzzling and mysterious that is commonly supposed, because it is essentially the same as: 1) the concept of ‘syntactic class’ in theoretical linguistics; and 2) as the concept of ‘data type’ in mathematics and computing.

8.1.1 The assumption that a wavefunction is a physical entity like a physical wave

The concept of a ‘wavefunction’ is described by Ball like this:

> "The French physicist Roland Omnès put it nicely when he called the wavefunction 'the fuel of a machine that manufactures probabilities'. In general, the chance of measuring any particular value of an observable property of a quantum system in an experiment can be calculated by a particular mathematical manipulation of its wavefunction. The wavefunction encodes this information, and quantum maths lets you extract it. There's a particular operation you conduct on the wavefunction to find a particle's momentum (*mass* × *velocity*), another operation to find its energy, and so on. In each case, what you get from this operation is *not* exactly the momentum, or energy, or whatever, that you'd measure in an experiment; it's the *average* value you'd expect to get from many such measurements." ([4, Locations 447–458], emphasis in the original).

However, a source of confusion in QM is when people believe that a 'wavefunction' describes a physical entity. In this connection, Ball quotes from an article by Berthold-Georg Englert:

> "[There is a] widespread habit of ... debaters to endow the mathematical symbols of the [Schrödinger] formalism with more meaning than they have. In particular, there is a shared desire to regard the Schrödinger wave function as a physical object itself after forgetting, or refusing to accept, that it is merely a mathematical tool that we use for a description of the physical object." [14, p. 12], quoted in [4, Locations 1398–1408].

Thus a wavefunction is not a physical entity, it is merely an abstraction:

> "The wave in Schrödinger's equation isn't a wave of electron charge density. In fact it's not a wave that corresponds to any concrete physical property. It is just a mathematical abstraction—for which reason it is not really a wave at all, but is called a *wavefunction*." ([4, Location 435], emphasis in the original).

8.1.2 The assumption that the 'collapse' of the wavefunction is a physical process that we might observe

Since a wavefunction is not a physical entity, it is also misleading to speak or write as if the 'collapse' of a wavefunction was a physical process that we might observe. In this connection, Freeman Dyson writes:

> "Unfortunately, people writing about quantum mechanics often use the phrase 'collapse of the wave function' to describe what happens when an object is observed. This phrase gives a misleading idea that the wave function itself is a physical object. A physical object can collapse when it bumps into an obstacle. But a wave function cannot be a physical object. A wave function is a description of a probability, and a probability is a statement of ignorance. Ignorance is not a physical object, and neither is a wave function. When new knowledge displaces ignorance, the wave function does not collapse; it merely becomes irrelevant." In "The Collapse Of The Wave Function" by Freeman Dyson in [9, p. 73].

8.1.3 The apparent misconception that quantum particles can be in more than one state at the same time

What appears to be another misconception about superposition is the idea that a quantum particle can be in two mutually exclusive states at the same time, as described by Ball:

> "The classical idea of a state generally has an exclusive aspect to it. Macroscopic objects can be a bit of this and a bit of that—a bit rigid but somewhat flexible, or kind of reddish brown. But they can't be in mutually exclusive states: *here* and *there*, having a mass of 1 g and also of 1 kg. I can't be cycling at 20 mph at the same time as cycling at 10 mph. And my cycling jacket can't be bright yellow at the same time as being pink. It can be a mixture of both, but it can't be all yellow and all pink. This seems common sense.
>
> "So it's understandable that, when we hear that quantum particles can be in more than one state at the same time, we struggle to see what that could mean, and we start to talk about quantum weirdness—or figure that we're too plain dumb to comprehend quantum mechanics." [4, Location 672].

And later he writes:

> "This 'two (or more) states at once' is called a superposition. The terminology conjures up the image of a ghostly double exposure. *But strictly speaking a superposition should be considered only as an abstract mathematical thing.* The expression comes from wave

> mechanics: we can write the equation for a wave as the sum of equations for two or more other waves.” ([4, Location 681], emphasis added).

One of the pioneers of QM, Paul Dirac, provides clarity:

> “The non-classical nature of the superposition process is brought out clearly if we consider the superposition of two states, A and B, such that there exists an observation which, when made on the system in state A, is certain to lead to one particular result, a say, and when made on the system in state B is certain to lead to some different result, b say. What will be the result of the observation when made on the system in the superposed state? The answer is that the result will be sometimes a and sometimes b, according to a probability law depending on the relative weights of A and B in the superposition process. It will never be different from both a and b. *The intermediate character of the state formed by superposition thus expresses itself through the probability of a particular result for an observation being intermediate between the corresponding probabilities for the original states, not through the result itself being intermediate between the corresponding results for the original states.*” ([12, Locations 359–367], emphasis in the original).

From what Ball and Dirac say, we can see that it is highly misleading for anyone to suggest that a quantum entity has two or more (mutually exclusive) states “at once” or “at the same time”.

As Ball makes clear in the second quote above, that way of describing a superposition of states is only valid if superposition is seen as an “abstract mathematical thing” with the implication that it is not a direct representation of any physical entity.

And Dirac describes how, in QM, a superposition of two mutually exclusive states, A, and B, are only superposed in the sense that they are *alternatives* to each other in a given context so that only A or B will yield a result when an observation is made.

This seems to get at the nub of the matter:

- *The word ‘superposition’ in wave mechanics* means two or more things (waves) *which can exist at the same time*—in the same way that someone may be tall *and* slim. But ...

- *The word 'superposition' in QM* is used to mean two or more things that are *mutually exclusive* in a given context, like the states A and B in the quote from Dirac, above.

We shall return to these points in Section 8.2.

8.1.4 The idea that measurable states of quantum particles do not have particular values until we measure them

With the double-slit experiment as the implied or explicit context for discussion, another thing which can be puzzling for many people about the Copenhagen concept of superposition, is the idea that the value of some variable only comes into existence by being measured or observed:

> "... we have no problem saying that [a] tennis ball was travelling at 100 mph and then I measured it. The tennis ball had the pre-existing property of a speed of 100 mph, which I could determine by measurement. We would never think of saying that it was travelling at 100 mph *because* I measured it. That wouldn't make any sense. In quantum theory, we do have to make statements like that. And then we can't help asking what it means. That's when the arguments start." [4, Locations 372–382], emphasis in the original).
>
> And later,
>
> "... for Bohr, all one can meaningfully say about a quantum system is contained in the Schrödinger equation. So if the maths says that we can't measure some observable quantity with more than a certain degree of precision, that quantity simply does not exist with greater precision. That is the difference between uncertainty ('I'm not sure what it is') and unknowability ('It *is* only to *this* degree'). ([4, Locations 1716–1725], emphasis in the original).

In terms of our everyday interpretation of things and events, the Bohr interpretation makes a certain amount of sense for such things as discoveries. Depending on one's use of language, a new discovery does not exist until the discovery has been made, although there is room for debate about the passage of time between the event itself and when reporters learn about it, and, from then, a further period of time before members of the public get to hear or read about it.

But more generally, the Copenhagen interpretation appears to be nonsense. For example, before we meet someone on a blind date (without any prior information), we are confident that he or she exists, and that he or she has a certain height, a certain hair colour, a certain timbre of voice, and so on, even though we do not know exactly what the values of those attributes may be. We see those values as things that exist prior to our meeting which we discover on the occasion of the meeting, not as values that spring into existence at the time we first see or hear them.

8.2 Possible remedies for problems of interpretation associated with quantum mechanics

It seems that, in Section 8.1.1 (about the concept of a 'wavefunction'), in 8.1.2 (about the 'collapse' of a wavefunction), and in 8.1.3 (about 'superposition' of states), the heart of the problem appears to be surprisingly widespread misunderstandings about QM. But in Section 8.1.4 (about the idea that the value of any state of a quantum particle does not exist until it is detected or measured), the problem seems to lie with QM itself.

The two subsections that follow describe similarities between the meaning of the word 'superposition' as it is used in QM (which, as previously noted, appears to be *different* from the meaning of that word in wave mechanics) and: 1) the concept of 'syntactic class' in theoretical linguistics; and 2) the concept of 'data type' in mathematics and computing.

8.2.1 The similarity between 'superposition' in QM and 'syntactic class' in theoretical linguistics

This section describes what seems to be a useful analogy for the idea of 'superposition' of two or more mutually exclusive states of a quantum entity (Section 8.1.3).

Figure 2 shows a set of SP-patterns that may be seen as a stochastic grammar[8] for a very simple English-like language.

The number at the end of each SP-pattern, preceded by a '$\star$', is a supposed frequency of occurrence of the SP-pattern in some imaginary text. Within every SPMA created by the SPS, the frequency of every participating SP-pattern is used to calculate absolute and relative probabilities for the SPMA, and inferences that may be drawn from the SPMA (Section 2.5).

The first SP-pattern in the figure, '`(S s1 D #D N #N V #V #S)`', represents the abstract structure of a sentence. Within that SP-pattern, '`D #D`' may be seen as a 'slot', 'space', or 'variable', for a word of the grammatical

---

[8] A 'stochastic' grammar is a grammar in which frequencies of usage of rules, or probabilities, have a role to play.

category ‘determiner’ (words like ‘the’, ‘a’, ‘two’, and so on), ‘`N #N`’ may be seen as a variable for a word of the category ‘noun’, and ‘`V #V`’ may be seen as a variable for a word of the category ‘verb’.

```
(S s1 D #D N #N V #V #S)*750
(D d1 the #D)*600
(D d2 this #D)*150
(N n1 dog #N)*400
(N n2 cat #N)*350
(V v1 walks #V)*500
(V v2 runs #V)*250
```

Figure 2: SP-patterns representing grammatical structures, as discussed in the text. The number after each SP-pattern, preceded by a ‘⋆’, is a supposed frequency of occurrence in some imaginary text.

The remaining SP-patterns in Figure 2 represent words, each one with its syntactic class:

- The SP-patterns ‘`(D d1 the #D)`’ and ‘`(D d2 this #D)`’ represent words of the syntactic class ‘determiner’ (marked by ‘`D`’ and ‘`#D`’ in the SP-pattern).
- The SP-patterns ‘`(N n1 dog #N)`’ and ‘`(N n2 cat #N)`’ represent words of the category ‘noun’.
- The SP-patterns ‘`(V v1 walks #V)`’ and ‘`(V v2 runs #V)`’ represent words of the category ‘verb’.

When the SPCM is run with the New SP-pattern, ‘`(this dog runs)`’, and with the SP-patterns in Figure 2 as Old SP-patterns, the best SPMA created by the program is the one shown in Figure 3. Here, “best” means that the SPMA in the figure is the one which, via an encoding (as described in [32, Section 4.1] and [30, Section 3.5]), yields the greatest compression of the New SP-pattern. Overall, the SPMA may be seen as a parsing of the sentence ‘`(this dog runs)`’ in terms of its grammatical constituents.

This example shows how, in a parsing of a simple sentence, each of the variables ‘`D #D`’, ‘`N #N`’, and ‘`V #V`’, may take on appropriate values. With other sentences, such as ‘`(the cat walks)`’ and so on, the variables would take on different values.

In summary, a superposition in QM is similar to a syntactic class in four respects:

- ‘Superposition’ in QM and ‘syntactic class’ in linguistics are both abstractions, without any corresponding physical structure in the world;

```
0           this         dog         runs       0
             |            |           |
1            |      N n1 dog #N       |         1
             |      |        |        |
2 S s1 D     |   #D N        #N V     |   #V #S 2
       |     |   |              |     |   |
3      D d2 this #D             |     |   |     3
                                |     |   |
4                               V v2 runs #V    4
```

Figure 3: The best SPMA created by the SPCM using the New SP-pattern '`(this dog runs)`' and the Old SP-patterns shown in Figure 2.

- They both represent two or more values that are *alternatives* in one or more contexts;
- With both superposition and syntactic class, there are situations where one value is selected out of the two or more values in the superposition or syntactic class. In QM, this happens when a 'particle' is detected or 'measured'. With syntactic classes, this happens when a 'variable' is assigned a value from the relevant syntactic class.
- In a superposition and in a syntactic class in a stochastic grammar, there is a probability or frequency associated with each value. Non-stochastic grammars in linguistics leave out this refinement, but in effect that means that all the values of any given syntactic class have the same probability or frequency.

8.2.2 The similarity between 'superposition' in quantum mechanics and 'data type' in mathematics and computing

As with superposition and syntactic class, there is a similarity between the concept of superposition in QM and the concept of a 'data type' in mathematics and computing. As before, it appears that they are similar in four respects:

- Both 'superposition' in QM and 'data type' in mathematics or computing are abstract constructs without any corresponding physical structure;
- A superposition and a 'data type' both represent a range of possible values that are alternatives in one or more contexts;
- In most applications, there are many situations where a 'variable' is assigned a specific value from the set of values in the data type;

- As with superposition, probabilistic programming on computers assigns a probability or frequency to each of the values of each data type. Where no such probabilities are assigned, we may assume that they are all the same.

8.3 Superposition and quantum computing

The two preceding subsections (Sections 8.2.1 and 8.2.2) seem to take a lot of the mystery out of the concept of superposition in QM. In the SP perspective, superposition is not some seemingly magical feature of QM but can be seen as a humdrum feature of theoretical linguistics and ordinary mathematics that may be modelled quite straightforwardly in the SPCM.

This section describes how these ideas relate to concepts in quantum computing. The main ideas in that area are described by Al-Khalili thus:

> "... in 1985, Oxford physicist David Deutsch published a pioneering paper that showed how [quantum computing] might be achieved in practice. ... Deutsch's machine would operate according to quantum principles to simulate any physical process. It required a row of quantum systems that could each exist in a superposition of two states, such as atoms in superpositions of two energy levels. These quantum systems would then be entangled together to create quantum logic gates that would be made to perform certain operations.
>
> The basic idea is that of the 'quantum bit' or qubit. In a normal digital computer, the basic component is the 'bit', a switch that can be in either of two positions: off or on. These are denoted by the binary symbols of 0 and 1. However, if a quantum system, such as an atom, is used then it could exist in the two states at once. A qubit can thus be both off and on at the same time, just as long as it can be kept isolated from its environment.
>
> Of course a single qubit is not very useful. But if we entangle two or more qubits we can start to see the power of such a set-up. Consider the information content of three classical bits. Each can be either 0 or 1 and so there are eight different combinations of the three (000, 001, 010, 100, 011, 101, 110, 111). But just three entangled qubits allow us to store all eight combinations at once! Each of the three digits is both a 1 and a 0 at the same time.
>
> Adding a fourth qubit would give us 16 combinations and a fifth, 32 and so on. The amount of information stored increases expo-

> nentially (as 2^N, where N is the number of qubits). Now imagine carrying out operations in the same way that we would with classical bits. We would be able to perform 2^N computations at once, the ultimate in parallel processing. Certain problems that might take a normal supercomputer years to solve could be cracked in a fraction of a second. [1, locations 3421–3433].

In the light of what is said about superposition by Ball and Dirac in Section 8.1.3, it seems necessary to view a qubit as an abstraction that describes ‘0’ and ‘1’ as alternatives in a given context. Contrary to the way qubits are often described for non-specialist readers, they are not versions of classical bits that can, in some mysterious way, be a “ghostly double exposure” [4, Location 681] of ‘0’ and ‘1’ at the same time.

8.3.1 Syntactic classes and quantum computing

If the parallels described in Sections 8.2.1 are accepted, then in the SP-pattern ‘`(S s1 D #D N #N V #V #S)`’ in Figure 2, the syntactic variable ‘`D #D`’ is like a superposition of the SP-patterns ‘`(D d1 the #D)`’ and ‘`(D d2 this #D)`’, the syntactic variable ‘`N #N`’ is like a superposition of the SP-patterns ‘`(N n1 dog #N)`’ and ‘`(N n2 cat #N)`’, and the syntactic variable ‘`V #V`’ is like a superposition of the SP-patterns ‘`(V v1 walks #V)`’ and ‘`(V v2 runs #V)`’.

Viewed in that way, the whole grammatical structure may be seen as having the same general form as the three entangled qubits in the quote from Al-Khalili near the beginning of Section 8.3 above: (000, 001, 010, 100, 011, 101, 110, 111). Like those three entangled qubits, the grammatical structure has the potential to create (or ‘generate’ in the jargon of theoretical linguistics) eight possible sentences like ‘`(the dog walks)`’, ‘`(the dog runs)`’, ‘`(the cat walks)`’, and so on, corresponding to the eight combinations, (000, 001, 010, 100, 011, 101, 110, 111).

8.3.2 Quantum computing in comparison with ordinary parallel processing

What is said about syntactic classes in Sections 8.2.1 suggests that, with a conventional computer working on a linguistic application, parallel processing may be applied very simply by treating each invocation of a syntactic class as an opportunity to process all the members of the class in parallel.

Since a syntactic class resembles a superposition in quantum computing (Section 8.3.1), we may suppose that much the same may be said about superpositions in quantum computing.

Thus, we may suppose that the anticipated speedup in any quantum computer would be largely because the '`0`' and '`1`' in each qubit would be processed in parallel, in essentially the same way as an ordinary parallel-processing computer, but with different hardware technologies.

That possibility, and several other possibilities are discussed quite fully by Ball in a chapter in [4] headed "Quantum computers don't necessarily perform 'many calculations at once' ". The possibilities described by Ball in several places between [4, Locations 3010], to [4, Location 3047].

In the light of these and other aspects of quantum computing, there are reasons to believe that ordinary non-quantum parallel processing may have advantages compared with quantum computing:

- The range of views described by Ball demonstrate many uncertainties in current thinking about quantum computing.

- The undoubted technical difficulties in making quantum computers work: "Just as in a classical computer, the 1s and 0s of the input to a quantum algorithm are marshalled into binary digits encoding solutions. The catch is that superpositions are generally very 'delicate'. They get easily disrupted by disturbances from the surrounding environment, particularly the randomizing effects of heat. ... this doesn't really mean—as is often implied—that superpositions are destroyed, but rather that the quantum coherence spreads into the environment, so that the original system decoheres." [4, Location 2822].

- It appears to be generally accepted that: "There isn't a straightforward way of making use of what QM has to offer, and designing good quantum algorithms is a very difficult task." [4, Location 3084]

- A report in the *Communications of the ACM* describes how a senior honors student, at the University of Texas at Austin, "discovered an algorithm that showed classical computers can indeed tackle predictive recommendations at a speed previously thought possible only with quantum computers." [16, p. 15].

- A paper by Mikhail Dyakonov [13] argues that the astronomically-large number of "degrees of freedom" in quantum computing means that, in answer to the question "When will we have a quantum computer?" in the title of the paper, "As soon as physicists and engineers learn to control this number of degrees of freedom, which means—never!" [13, p. 4].

- In short, it seems likely that any speed advantage of a quantum computer compared with an ordinary parallel-processing computer, would be due to the underlying hardware rather than QM concepts as such.

In view of these considerations, it seems that there is a case for devoting at least as much effort to the development of non-quantum parallel processing as is currently devoted to the development of quantum computing:

- It seems as likely as not that, in quantum computers, gains in the anticipated speed of processing, or reductions in computational complexity, could be matched with non-quantum parallel processing applied to knowledge structures with implicit parallelism like syntactic classes and mathematical data types.
- If it turns out that there is something special about bit-level parallel processing, that kind of parallelism could probably be developed with non-quantum computers.
- It seems likely that engineering problems in the advancement of non-quantum computers would be more easily solved than problems such as decoherence that have proved so hard to solve with quantum computers.
- It may be that automatic, semi-automatic, or manual programming of non-quantum computers could prove to be more straightforward than programming quantum computers.

9 Nonlocality, entanglement, SP-multiple-alignment, and discontinuous dependencies

The interrelated concepts of 'superposition' (discussed in Section 8), 'nonlocality' and 'entanglement' are described by Al-Khalili thus:

> "[Superposition] states that a quantum particle can be in a combination of two or more (mutually exclusive) states at the same time, while [nonlocality] says that two quantum particles (or two separate parts of the spread-out wavefunction of the same particle) can somehow remain in touch with each other however far apart they are. I will now combine these two ideas together in order to introduce a third quantum concept.
>
> "In quantum mechanics, the idea of two dice remaining in (nonlocal) contact with each other how ever far apart they are is known as entanglement." [1, 1225–1234].

There is little doubt that the phenomena of "nonlocality" and "entanglement" are genuine features of the world and not merely some "weirdness" in QM which may, at some stage, be explained away:

> "Today, quantum nonlocality and entanglement are no longer the subject of philosophical debate. They are accepted as crucial features of the quantum world. Indeed, entanglement of many particles could lead to the development of a whole new technology not even dreamed of by the quantum pioneers." [1, location 1274].

9.1 How the concepts of nonlocality and entanglement may be represented via the concept of SP-multiple-alignment in the SP Theory

As with superposition (Section 8), there is a potentially useful analogy for nonlocality and entanglement in the processing of natural language. This may be seen as another example of the way aspects of cognitive psychology may have a bearing on the development of theories in physics (Section **??**).

The SPMA shown in Figure 5 provides an example (with simplifications of some of the details of English grammar). Here, the sentence '`t w o k i t t e n s p l a y`' is identified as a sentence (defined by the SP-pattern '`(S ... #S)`' in row 7), and parsed into constituents such as a noun phrase ('`(NP ... #NP)`' in row 4), a verb phrase ('`(V ... #V)`' in row 6), and so on.[9]

For present purposes, the key point of interest is that, within sentences like this, there is a syntactic "dependency" between the 'subject' at the beginning (which is the noun-phrase '`t w o k i t t e n s`') and the main verb-phrase later in the sentence (which is the single word '`p l a y`').

The rule here is that, in English at least, if the 'subject' noun-phrase is plural then the main or only verb-phrase must be plural, and if the subject noun-phrase is singular then the main verb-phrase must be singular. Most natural languages have dependencies like that, such as for example, gender dependencies in French, which may cut across number dependencies (for more discussion, see [30, Section 5.4]).

This kind of dependency is often described as "discontinuous" because it can jump over intervening structure such as *that were only born yesterday*

---

[9] It is anticipated that, when the SPS is more fully developed (Section 3.2), it will be able, via unsupervised learning, to discover Old patterns like those in rows 1 to 8 of Figure 5 for itself, instead of them being provided ready-made in the store of Old patterns as is the case with that example.

between the subject noun-phrase, *two kittens*, and the main verb-phrase, *play*, in *two kittens that were only born yesterday play*, and there appears to be no limit on how big that intervening structure may be.

Amongst the several ways in which discontinuous dependencies may be represented in AI systems, one of the simplest is via an SP-pattern within an SPMA, like the SP-patten in row 8 in Figure 5. Here, the SP-symbol '`NPp`' (mnemonic for 'plural noun-phrase') is aligned with a matching SP-symbol within the SP-pattern for the subject noun-phrase, '`(NP NPp D Dp #D N Np #N #NP)`' in row 4, and the symbol '`VPp`' (mnemonic for 'plural verb-phrase') is aligned with a matching symbol within the SP-pattern for the main verb-phrase, '`(VP VPp Vr #Vr #VP)` in row 6.

The fact that the SP-symbols '`NPp`' and '`VPp`' both appear in one SP-pattern (in row 8) is what marks the dependency between the subject noun-phrase and the main verb-phrase.

This example suggests that insights gained with the SPS may have traction in QM. It seems possible that a dependency between, for example, two entangled electrons, such that one electron has a clockwise spin while the other electron has a counter-clockwise spin, may be understood in a manner that is similar to our understanding of the phenomenon of syntactic dependencies in natural languages. In both cases:

- There is a correlation between the two elements of the dependency.
- The dependency may bridge arbitrarily large amounts of intervening structure.
- There is a kind of 'instant' communication in the sense that, if we know one element of a dependent pair, we know immediately what the other should be. This effect is what Einstein famously called 'spooky action at a distance'.[10]

The kind of instant communication just mentioned—something that has been verified in many experiments—looks like communication that is faster than the speed of light and thus incompatible with a basic principle in general relativity that nothing can travel faster than light. How can that contradiction be resolved?

---

[10] "As he declared to his friend Max Born, coining a memorable phrase, 'Physics should represent a reality in time and space, free from spooky action at a distance.'", this quote, including the quote from Einstein, is in [20, Location 8066]. The source for what Einstein said is given as "Einstein to Max Born, March 3rd, 1947, in [8, p. 155] (not in *Albert Einstein Archives*).

9.2 How a non-local, entangled pair of particles may be regarded as a single entity

A suggested answer to the question at the end of the preceding subsection is that what is normally construed as *two* entangled particles could equally equally well be seen as a single object, in the same way that the SP-pattern '`(Num PL ; NPp VPp)`' is a single object containing the two significant SP-symbols, '`NPp`' and '`VPp`'. In this case, there is no need for any communication at all, spooky or otherwise, because if we have a full knowledge of an SP-pattern, we know its contents.

Ball makes essentially the same point in a chapter in [4] headed "There is no 'spooky action at a distance' " [4, location 1973]. Without attempting to discuss all the arguments and counter-arguments that Ball considers, here is one of the more telling examples that he describes:

> "Think of a pair of gloves: one left-handed, the other right-handed. If we were to post one at random to Alice in Aberdeen and the other to Bob in Beijing ..., then the moment Alice opened the parcel and found the left glove (say), she'd know that Bob's glove is right-handed. This is trivial, because the gloves had that handedness all the time they were in transit—it's just that Alice and Bob didn't know which was which until one of them looked." [4, locations 1838–1845].

Here, the pair of gloves may be seen as a discrete entity rather than two separate entities. This is like the SP-pattern '`(Num PL ; NPp VPp)`' in Figure 5 being regarded as a single entity which provides knowledge of both '`NPp`' and '`VPp`' without the need for communication between them.

In the same vein, a little later, Ball writes that:

> "We can't regard particle A and particle B [that are entangled] as separate entities, even though they are separated in space. As far as quantum mechanics is concerned, entanglement makes them both parts of a single object." [4, location 2026].

Here's another example. Imagine a scene in which a car is partly obscured by the trunk of a tree, with the front part visible. If we see the front part move forwards (or backwards), we can infer instantly that the back of the car will be moving in the same direction and at the same speed. Of course, it could be a stage magician's car that does something different—so, for that reason, the inference is probabilistic. But in this case the probabilities strongly favour the normal interpretation.

Since this kind of scene is very familiar, it would be strange indeed if people were to speak of nonlocality and entanglement between the front and back of a car! Perhaps we'll eventually drop that kind of language when speaking or writing about entangled quantum particles.

Similar things can be said about left and right brackets as they are normally used in text. Although they are normally separated by a body of text, which can be quite variable in its size, we think of them as belonging to a single entity, which leads to expectations, such as a left curly bracket being followed by a right curly bracket, or a left square bracket being followed by a right square bracket, and so on.

10 Conclusions

Acknowledgements

I am grateful to anonymous referees for constructive comments on earlier versions of this paper.

A Outline of the SPS

The *SP System* (SPS), meaning the *SP Theory of Intelligence* and its realisation in the *SP Computer Model* (SPCM), is the product of a lengthy programme of research, *seeking to simplify and integrate observations and concepts across artificial intelligence (AI), mainstream computing, mathematics, and HLPC.*

Although this objective is ambitious, the research has met with success in the discovery and development of the concept of the *SP-multiple-alignment* (SPMA) concept within the SPS (Appendix A.5), and the exploration of how it may function in the modelling of different aspects of intelligence, and writing a book about the research [30]. That process, including the development and testing of *hundreds* of versions of the SPMA within the CPCM, has taken about 17 years, and there has been a further period of research, exploring potential benefits and applications of the SPCM [42].

Because of substantial evidence for IC as a unifying principle in brains and nervous systems [40], a working hypothesis in the SP programme of research is that IC would be central in the structure and workings of the SPS.

The SPS and some of its applications are described most fully in the book *Unifying Computing and Cognition* [30], more briefly but quite fully in the paper [32], and in outline in this Appendix. Since the SP concepts provide a foundation for the proposals and discussion in this paper, readers may find it useful to read at least one of those descriptions.

Mathematical underpinnings of the SPCM are described in Appendix D.

Key publications in this programme of research, including several about potential applications of the SPS, are detailed with download links on bit.ly/37Y0NcI. A more comprehensive list is on on bit.ly/2Gxici2.

Because the SPS is the product of a lengthy programme of research, seeking to simplify and integrate observations and concepts across a broad canvass (noted at the beginning of this Appendix A), it naturally has points of resemblance to many other systems. This has sometimes been construed, quite wrongly, to mean that the SPS is "nothing but X", or "nothing but Y", etc. With the aim of reducing the chances of misunderstandings in this area, distinctive features and advantages of the SPS are described in [38].

To forestall any misunderstandings, it cannot be emphasised too strongly that the SPS is *radically* different from deep neural networks (DNNs) [38, Section V].

The organisation and workings of the SPS are outlined in the subsections that follow.

A.1 The SPS as a brain-like system

The SPS is conceived as a brain-like system that receives *New* information that is not compressed from its environment via its senses and stores some or all of it in compressed form in its 'brain' as *Old* information. This is illustrated schematically in Figure 4.

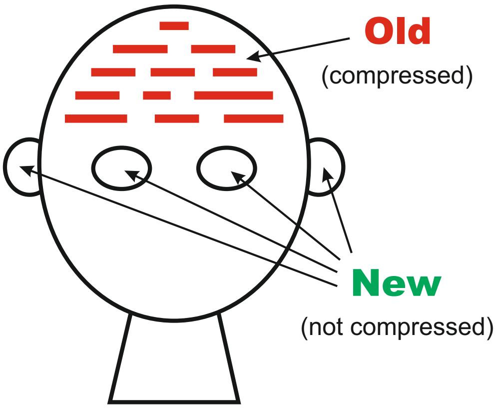

Figure 4: A schematic representation of the SPS. Adapted from Figure 1 in [32], with permission.

A.2 The central importance of information compression

In the SPS, IC is a unifying principle because of the previously-mentioned evidence for the importance of IC in HLPC (Appendix **??**). More specifically, a central idea in the SPS is that IC may be achieved via the matching and unification of patterns (Appendix **??**) which itself may be achieved via the creation of SPMAs (Appendix A.5).

A.3 The SP Computer Model

The SP Computer Model (SPCM) is a computer program which gives expression to the elements of the SP Theory of Intelligence.

As previously noted, Appendix D describes the mathematical underpinnings of SPCM, both the mathematics included in the model and mathematics that has contributed to its development.

The development of the SPCM proceeded hand-in-hand with the development of the SP Theory. The aim, as mentioned above, has been to discover a framework for the simplification and integration of observations and concepts across a broad canvass.

From the germ of an idea about how this may be done, the process of creating a conceptual framework and developing it was done via the development and testing of a very large number of versions of the SPCM. At all stages, simplification and integration of observations and concepts has been the touchstone of success or failure.

The lengthy process of development has been extremely important in weeding out bad ideas and blind alleys in the evolving SP Theory, and it is largely responsible for the intelligence-related strengths and potential of the SPS, summarised in Appendix A.7. It is also largely responsible for the several potential benefits and applications of the SPS, summarised in Appendix A.8.

Thus the SPCM was not hacked together in a day. Its long period of research, development, and testing, provides a solid foundation for the SP Theory of Intelligence.

Source code and Windows executable code for the SPCM may be downloaded via links from www.cognitionresearch.org/sp.htm#SOURCE-CODE.

A.4 SP-patterns and SP-symbols

All information in the SPS is stored and processed as *SP-patterns*, where an SP-pattern is an array of atomic *SP-symbols* in one or two dimensions. An

SP-symbol is simply a mark (from an alphabet of marks) that can be matched with any other SP-symbol in an all-or-nothing manner. One-dimensional SP-patterns are normally shown with round brackets (‘`(`’ and ‘`)`’) at each end.

At present, the SPCM works only with one-dimensional SP-patterns. It is envisaged that it will be generalised to work also with two-dimensional SP-patterns which should open the door to the representation and processing of pictures, diagrams, and the like.

It would also facilitate the learning of an object or other structure in three dimensions as described in [33, Sections 6.1 and 6.2]. The basic idea is to take overlapping pictures of the object from several different angles and then knit them together by unifying the overlapping areas, in much the same way that a panoramic picture of a scene can be created from several overlapping pictures. The technique for creating 3D digital models of objects is now offered as a service by several commercial companies.

The introduction of two-dimensional SP-patterns would also facilitate the representation of procedural knowledge with parallel processing [34, Section IV-H].

A.5 SP-multiple-alignment

Compression of information in the SPS is achieved largely via the building of SPMAs, a powerful concept which has been borrowed and adapted from the concept of ‘multiple sequence alignment’ in bioinformatics. An example of an SPMA is shown in Figure 5.

```
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 NPp D Dp         #D N Np                        #N #NP             | | | |            4
             |   |                                                  |              | | | |
5            |   |                                                  |         Vr 1 p l a y #Vr        5
             |   |                                                  |         |             |
6            |   |                                                  |  VP VPp Vr           #Vr #VP    6
             |   |                                                  |  |   |                    |
7 S Num    ; NP  |                                                 #NP VP  |                   #VP #S 7
     |     |     |                                                         |
8   Num PL ;    NPp                                                       VPp                         8
```

Figure 5: The best SPMA created by the SPCM with a store of Old SP-patterns like those in rows 1 to 8 (representing grammatical structures, including words) and a New SP-pattern, ‘`(t w o k i t t e n s p l a y)`’, shown in row 0 (representing a sentence to be parsed). Adapted from Figure 1 in [31], with permission.

The concept of SPMA is central in the workings of the SPS and provides most of the AI-related versatility of the SPS, summarised in Appendix A.7, below.

Bearing in mind that underselling concepts is as bad as overselling them, *the concept of SPMA has the potential to be as significant for an understanding of 'intelligence' as is DNA for biological sciences. It may prove to be the "double helix" of intelligence.*

A.5.1 How IC is achieved via SPMAs

An SPMA provides for the economical encoding of a New SP-pattern in row 0 (sometimes more than one) in terms of one or more Old SP-patterns, one per row, in the other rows of the SPMA. For a given SPMA, the amount of compression of the New SP-pattern that is achieved is calculated as described in [32, Section 4] and [30, Sections 3.4 and 3.5].

In Section 2.2.9, there is an outline of seven techniques for IC in the SPS, the last and most powerful of which is the SPMA concept.

In Appendix B, there is a detailed description of how the SPMA concept may model any or all of the other six techniques for IC, in any combination.

A.5.2 Heuristic search in the building of SPMAs

Because the abstract space of possible SPMAs is astronomically large, it is not possible to search it exhaustively. Thus it is necessary use heuristic search: building SPMAs in stages, and retaining only the best ones at the end of each stage. Here, 'best' means the SPMAs that can provide relatively high levels of IC.

Heuristic search in the SPCM trades accuracy for speed, allowing SPMAs to be built in a reasonable time, but without any guarantee that best possible SPMA has been found. However, the search process normally produces SPMAs that are good enough for how they are to be used.

A.5.3 SPMAs, inference, and probabilities

Because of the intimate relation between IC and concepts of inference and probability (Appendix **??**), each SPMA has associated probabilities ([32, Section 4.4], [30, Section 3.7]). For each SPMA, there are: 1) the *absolute probability* of the encoded version of the New SP-pattern, and 2) the very much more useful *relative probability* which facilitates the comparison of one SPMA with another. Also, for such things as probabilistic reasoning ([32, Section 10], [30, Chapter 7]), the SPCM can, with two or more SPMAs, calculate the relative probabilities of SP-patterns and SP-symbols ([32, Section 4.4.4], [30, Section 3.7.3]).

A.5.4 An abstract view of the SPMA concept in terms of patterns of redundancy

From an abstract perspective, row 0 of an SPMA may be seen as raw data from the system's environment (represented by a New SP-pattern), and each of the rows below row 0 (which contains one Old SP-pattern) may be seen as a pattern of redundancy in the world (which, in a more fully developed SPCM, the system would normally learn via its procedures for unsupervised learning).

Since each SPMA may always be collapsed into a single sequence, with the alignments between different rows showing the relative positions of different patterns of redundancy, an SPMA may be seen as a single pattern of redundancy. Along the length of that pattern, each part that is matched to a high-frequency Old SP-pattern may be seen to represent a high level of redundancy, each part that is matched to a low-frequency Old SP-pattern may be seen to represent a low level of redundancy, and there may be many values in between.

Since there is in principle no limit to the number of Old SP-patterns that may appear in an SPMA, or their frequencies of occurrence, there is in principle no limit to the number of patterns of redundancy that may be included within the SPMA structure. For that reason, there is in principle no limit to the accuracy with which that SPMA structure reflects the structure of the world.

A.6 Unsupervised learning

Unsupervised learning in the SPS means the intake of New information and its storage in compressed form as Old information, as outlined in Appendix A.1, above. Compression of information is achieved both via the creation of SPMAs, and via the creation of *SP-grammars*, where an SP-grammar is a set of SP-patterns that have been shown to be effective, collectively, in the compression of a given body of New information. How much compression may be achieved via a given SP-grammar is calculated as described in [32, Section 5] and [30, Chapter 9].

An SP-grammar may be seen, not only as a collection of SP-patterns, but also as a computational model of the New information from which it was derived. It is anticipated that unsupervised learning may encompass the creation of three-dimensional computational models, as described in Appendix A.4.

A.7 Intelligence-related strengths and potential of the SPS

The versatility and potential of the SPS in intelligence-related features are summarised in [41, Section 3.7], and that summary is reproduced in the following four subsections.

A.7.1 *Versatility in intelligence-related capabilities*

The SPS has strengths and potential in several 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 subsection); planning; and problem solving. There is more detail in [32] and [30].

A.7.2 *Versatility in reasoning*

The strengths and potential of the SPS in reasoning include: 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; a non-Bayesian alternative to 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 [32, Section 10] and [30, Chapter 7]. There is also potential for spatial reasoning [34, Section IV-F.1] and what-if reasoning [34, Section IV-F.2].

A.7.3 *Versatility in the representation of diverse kinds of intelligence-related knowledge knowledge*

Because of the expressive power of the SPMA concept (Appendix A.5), the SPS has strengths and potential in the representation and processing of several kinds of intelligence-related 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 'data type'. The addition of two-dimensional SP-patterns to the SPCM is likely to expand the representational repertoire of the SPS to structures in two- and three-dimensions and the representation of procedural knowledge with parallel processing (Appendix A.4).

A.7.4 *Seamless integration of intelligence-related capabilities and knowledge*

Because of the versatility of the SPS outlined in Sections A.7.1, A.7.2, and A.7.3, outlined above, and because this versatility is largely due to the central role of SPMA, there is clear potential for the seamless integration of diverse intelligence-related capabilities and diverse kinds of intelligence-related knowledge, in any combination.

It appears that that kind of versatility and seamless integration, illustrated schematically in Figure 6, is *essential* in any artificial system that aspires to human-level broad AI, with fluidity across diverse aspects of intelligence, and with adaptability in the acquisition of new skills, new knowledge, and new ideas.

A.8 Other potential benefits and applications of the SPS

The paper [36] outlines of some of the potential benefits and applications of the SPS, apart from those outlined in Section A.7. These other potential benefits and applications are described at more length in other papers: how the SPS may facilitate the management of big data [35]; how the system may help in the development of intelligence in autonomous robots [34]; how it may help in the understanding of natural vision and the development of computer vision [33]; how it may function as an intelligent database system [31]; and how it may help in the processes of medical diagnosis [29].

A.9 Potential to solve problems in AI research

.

The SPS has clear potential to help solve 19 problems in AI research [?]. All but 2 of those problems have been described by leading researchers in AI, in interviews with the writer Martin Ford and presented in his book *Architects of Intelligence* [15].

A.10 A foundation for the development of human-like general AI

The strengths and potential of the SPS described in Appendices A.7 to A.9 suggest that *the SPS provides a firmer foundation for the development of AGI than any alternative*, and that includes DNNs.

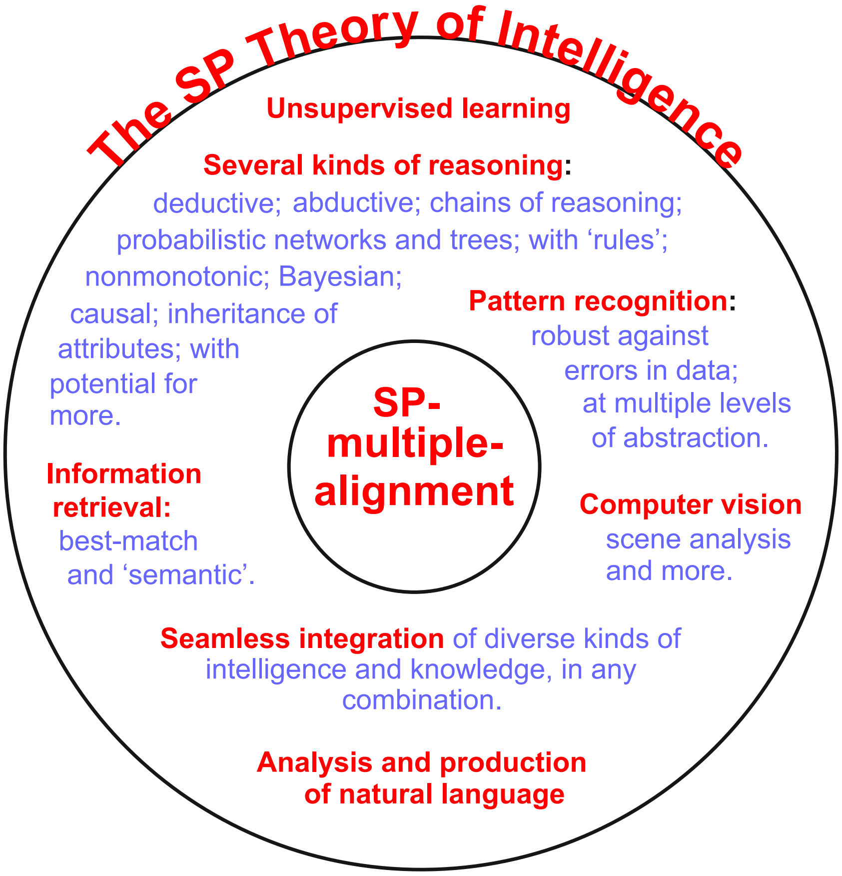

Figure 6: A schematic representation of versatility and integration in the SPS, with SPMA at centre stage. Adapted from Figure 6 in [40], with permission.

A.11 SP-Neural

SP-Neural is a version of the SPS in which concepts such as SP-pattern, SP-symbol, and SPMA are expressed in terms of neurons with their interconnections and intercommunications [37].

A.12 SP Machine

It is intended that the 'SP Machine' will be derived from the SPCM with the application of high levels of parallel processing provided by one or more GPUs with many cores, and improvements in its user interface.

It is envisaged that the SP Machine will be a vehicle for further research and development, by individual researchers and groups, towards an industrial-strength device, with guidance from a 'roadmap' presented in [25]. This development is shown schematically in Figure 7.

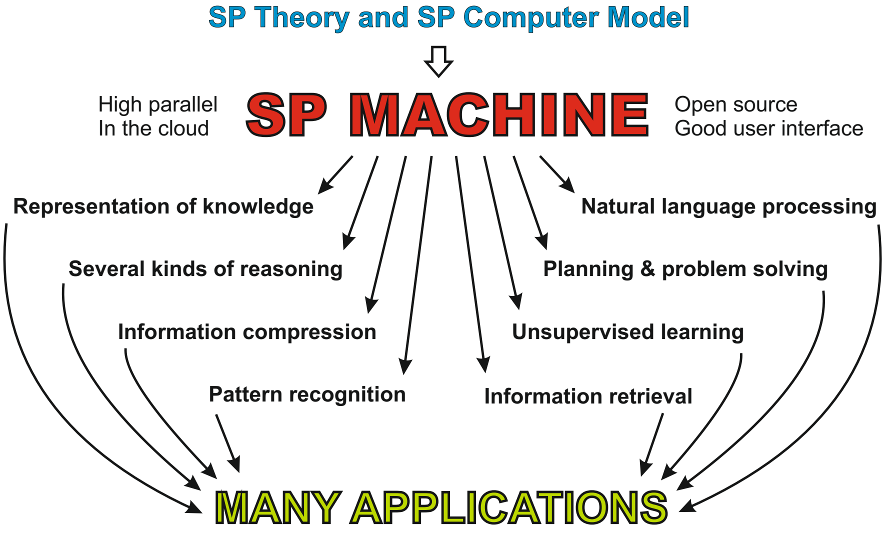

Figure 7: Schematic representation of the development and application of the SP machine. Reproduced from Figure 2 in [32], with permission.

Since the SPS is part of the proposed NM, the SP Machine may also provide a basis for the development of the NM.

B How each of six variants of ICMUP may be seen as special cases of the SP-multiple-alignment concept

In writing about the SP system, it has proved useful in Section 2.2 to outline seven variants of ICMUP.

The purpose of this appendix is to describe with examples how the first six of those variants of ICMUP may be realised via the SPMA concept, how each of those six variants of ICMUP should be seen as instances of the more general principle of information compression via SPMA, and how all six variants may be integrated seamlessly within the SPMA framework, in any combination.

Most of this appendix has been transferred from the unpublished technical report [39], with editing. All references to this material should now be made to this appendix, not the earlier technical report.

B.1 Basic ICMUP

The simplest version of ICMUP is where two patterns match each other and they are merged or 'unified' to make one. This may be modelled very simply and directly in the SPMA framework with a simple match between all or part of a New SP-pattern (such as '`k i t t e n`' in Figure 5) and one Old SP-pattern (such as '`Nr 5 k i t t e n #Nr`' in the same figure).

B.2 Chunking-with-codes

The basic idea here is that, with each unified 'chunk' of information, give it a relatively short name, identifier, or 'code', and use that as a shorthand for the chunk of information wherever it occurs. This is illustrated by the same example as was given in Appendix B.1. Here, '`k i t t e n`' is the chunk of information and the SP-symbols '`Nr 5 ... #Nr`' serve as the code for that chunk.

B.3 Schema-plus-correction

The schema-plus-correction may be illustrated with a simple example: the kinds of things that need to be done in serving a meal in a restaurant or cafe in response to an order from a customer. The example will be described first, and how it illustrates the schema-plus-correction concept will follow.

Figure 8 shows an *SP-grammar*, comprising a collection of SP-patterns, which may be seen as a highly simplified program for the serving of a meal to order.

The first SP-pattern in the figure, ‘`PM ST #ST MC #MC PD #PD #PM`’ describes the overall structure of the procedure for serving a meal. It is identified by the pair of SP-symbols ‘`SM ... #SM`’, mnemonic for ‘serve meal’.

The main steps are the serving of a starter (‘`ST ... #ST`’), the serving of the main course (‘`MC ... #MC`’), and the serving of a pudding (‘`PD ... #PD`’). Corresponding SP-patterns are shown in the second and subsequent rows in the figure: there are three kinds of starter, five kinds of main course, and four kinds of pudding.

```
SM ST #ST MC #MC PD #PD #SM  | Serve meal
ST 0 mussels #ST             | Starter: serve a dish of mussels
ST 1 soup #ST                | Starter: serve a bowl of soup
ST 2 avocado #ST             | Starter: serve an avocado dish
MC 0 lassagne #MC            | Main course: serve a lassagne dish
MC 1 beef #MC                | Main course: serve a beef dish
MC 2 nut-roast #MC           | Main course: serve a nut-roast dish
MC 3 kipper #MC              | Main course: serve a kipper
MC 4 salad #MC               | Main course: serve a salad
PD 0 ice cream #PD           | Pudding: serve ice cream
PD 1 apple crumble #PD       | Pudding: serve apple crumble
PD 2 fresh fruit #PD         | Pudding: serve fresh fruit
PD 3 tiramisu #PD            | Pudding: serve tiramisu
```

Figure 8: An SP-grammar comprising a set of SP-patterns representing, in a highly simplified form, the kinds of procedures involved in serving a mean for a customer in a restaurant or cafe. To the right of each SP-pattern is an explanatory comment, after the symbol ‘`|`’.

To see how this grammar functions in practice, consider the SPMA shown in Figure 9.

This SPMA is the best one created by the SPCM with the New SP-pattern, ‘`SM 0 4 1 #SM`’, processed in conjunction with Old SP-patterns shown in Figure 8. Here, the New SP-pattern may be seen as an economical description of what the customer ordered: a starter comprising a dish of mussels represented by the short code ‘`0`’; a main course chosen to be a salad represented by the short code ‘`4`’; and a pudding which in this case is apple crumble, represented by the code ‘`1`’.

Assuming that each of the SP-symbols ‘`mussels`’, ‘`salad`’, and ‘`apple crumble`’, represents the execution of instructions for serving the corresponding dish, the whole SPMA may be seen to achieve the effect of serving what the customer has ordered.

```
0     1     2       3         4

SM -- SM                               | Serve meal
      ST ---------- ST                 | Serve starter
0 ----------------- 0
                    mussels
      #ST --------- #ST
      MC -- MC                         | Serve main course
4 --------- 4
            salad
      #MC - #MC
      PD -------------------- PD       | Serve pudding
1 --------------------------- 1
                              apple
                              crumble
      #PD ------------------- #PD
#SM - #SM

0     1     2       3         4
```

Figure 9: The best SPMA created by the SPCM with the New SP-pattern, '`SM 0 4 1 #SM`', and the set of Old SP-patterns shown in Figure 8. Comments are shown on the right, each one following the symbol '`|`'.

In terms of the schema-plus-correction concept, the SP-pattern '`SM ST #ST MC #MC PD #PD #SM`' may be seen as the schema, which in this case is the overall structure of the meal. Like any other schema, it may also be seen as a chunk of information which in this case has the code '`SM ... #SM`'.

In this example, the 'corrections' to the schema are the choices made by the customer, represented by the digits in the SP-pattern '`SM 0 4 1 #SM`'. As noted above, '`0`' means "Serve the starter mussels", '`4`' means (serve) salad, and '`1`' means (serve) apple crumble.

Although the example is very simple, it shows how a meal can be described very economically with the SP-pattern '`SM 0 4 1 #SM`'. Because the menu has been identified by the pair of symbols '`SM`' and '`#SM`', it is not necessary to put in information to identify the starter, main course, and pudding, or any other information that is associated with those three courses.

B.4 Run-length coding

The variant called 'run-length coding' may be used with any sequence of two or more copies of an SP-pattern. In that case, it is only necessary to record one copy of the SP-pattern, with something to define the length of the sequence, such as the number of copies of the SP-pattern, or tags to mark the start and end of the sequence.

With the SP system, run-length coding may be achieved via recursion, as described in Section B.4.1.

B.4.1 Recursion

The SP system does not provide for the repetition of procedures via these kinds of statement: *while ...*, *do ... while ...*, *for ...*, or *repeat ... until ...*. But the same effect may be achieved via recursion, as illustrated in Figure 10.

```
0     1             2     3             4             5     6             7     8             9     10

pg ---------------- pg
                    a ----------------------------------------------------------------------------- a
a6 ------------------------------------------------------------------------------------------------ a6
                                                                                                    procedure-A
                    #a ---------------------------------------------------------------------------- #a
                    ri ------------------------------ ri
                                                      ri1
                                                      ri ---------------- ri
                                                                          ri1
                                                                          ri ---------------- ri
                                                                                              ri1
                                                                                              ri
                                                                                              #ri
                                                                                b ----------- b
b1 ---------------------------------------------------------------------------- b1
                                                                                procedure-B
                                                                                #b ---------- #b
                                                                          #ri --------------- #ri
                                                            b ----------- b
b1 -------------------------------------------------------- b1
                                                            procedure-B
                                                            #b ---------- #b
                                                      #ri --------------- #ri
                                        b ----------- b
b1 ------------------------------------ b1
                                        procedure-B
                                        #b ---------- #b
                    #ri ----------------------------- #ri
      c ----------- c
c4 -- c4
      procedure-C
      #c ---------- #c
                    d --- d
d3 ---------------------- d3
                          procedure-D
                    #d -- #d
#pg --------------- #pg

0     1             2     3             4             5     6             7     8             9     10
```

Figure 10: An SPMA produced by the SP model showing how, via recursion mediated by a self-referential SP-pattern (which is '`ri ri1 ri #ri b #b #ri`' in this example), the SP system may model the repetition of a procedure or function, which in this examples is shown as '`procedureB`'.

A key point here is that the SP-pattern '`ri ri1 ri #ri b #b #ri`' (which appears in columns 5, 7, and 9) is recursive because it is self-referential, and this is because the pair of SP-symbols '`ri #ri`' within the larger SP-pattern '`ri ri1 ri #ri b #b #ri`' may be matched and unified with the same two SP-symbols at the beginning and end of that larger SP-pattern. Hence, the larger SP-pattern contains a reference to itself.

B.5 Class-inclusion hierarchies with inheritance of attributes

The way in which a class-inclusion hierarchy may be modelled in the SP system is described in Section B.5.1, with an illustration in Figure 11.

B.5.1 Example: a class-inclusion hierarchy in the SP system

In Figure 11, the SPMA produced by the SPCM shows how a previously unknown entity with features shown in the New SP-pattern in column 1 may be recognised at several levels of abstraction: as an animal (column 1), as a mammal (column 2), as a cat (column 3) and as the specific cat 'Tibs' (column 4).

From this SPMA, we can see how the entity that has now been recognised *inherits* unseen characteristics from each of the levels in the class hierarchy: as an animal (column 1) the creature '`breathes`' and '`has-senses`'; as a mammal it is '`warm-blooded`'; as a cat it has '`carnassial-teeth`' and '`retractile-claws`'; and as the individual cat Tibs it has a '`white-bib`' and is '`tabby`'.

B.6 Part-whole hierarchies with inheritance of contexts

This is like class-inclusion hierarchies with inheritance of attributes except that the structure represents the parts and subparts of some entity. Each subpart may be seen to inherit its place in larger structures.

The way in which a part-whole hierarchy may be modelled in the SP system is described in Section B.6.1.

B.6.1 Example: a part-whole hierarchy in the SP system

Figure 12 shows how a part-whole hierarchy may be accommodated in the SP system. Here, an SP-pattern representing the concept of a car is shown in column 2, with parts such as '`<engine>`' '`<body>`', and '`<gearbox>`'. The SP-pattern in column 1 shows parts of an engine such as '`<cylinder-block>`', '`<pistons>`', and '`<crankshaft>`'. The SP-pattern in column 3 shows how the body may be divided into such things as '`<steering-wheel>`', '`<dashboard>`', and '`<seats>`'. The SP-pattern in column 5 divides the dashboard into parts that include '`<speedometer>`' and '`<fuel-gauge>`'. And the SP-pattern in column 4 divides the speedometer into '`<speed-dial>`', '`<speed-pointer>`', and more.

```
0           1            2              3                  4

                                                           T
                                                           Tibs
                                        C ---------------- C
                                        cat
                         M ------------ M
                         mammal
            A ---------- A
            animal
            head ---------------------- head
                                        carnassial-teeth
            #head --------------------- #head
            body ----------------------------------------- body
white-bib ------------------------------------------------ white-bib
            #body ---------------------------------------- #body
            legs ---------------------- legs
                                        retractile-claws
            #legs --------------------- #legs
eats ------ eats
            breathes
            has-senses
            ...
            #A --------- #A
furry ------------------ furry
                         warm-blooded
                         ...
                         #M ----------- #M
purrs --------------------------------- purrs
                                        ...
                                        #C --------------- #C
                                                           tabby
                                                           ...
                                                           #T

0           1            2              3                  4
```

Figure 11: The best SPMA found by the SP model, with the New SP-pattern '`white-bib eats furry purrs`' shown in column 1, and a set of Old SP-patterns representing different categories of animal and their attributes shown in columns 1 to 4. Reproduced with permission from Figure 6.7 in [30].

```
0                  1                  2           3                  4                 5

                                      <car>
                                      c1
                   <engine> --------- <engine>
                   e1
                   <cylinder-block>
                   <pistons>
<crankshaft> ----- <crankshaft>
                   ...
                   </engine> -------- </engine>
                                      <body> ---- <body>
                                                  b1
<steering-wheel> -------------------------------- <steering-wheel>
                                                  <dashboard> ------------------------ <dashboard>
                                                                                       d1
                                                                     <speedometer> --- <speedometer>
                                                                     sp1
<speed-dial> ------------------------------------------------------- <speed-dial>
                                                                     <speed-pointer>
                                                                     ...
                                                                     </speedometer> -- </speedometer>
                                                                                       <fuel-gauge>
                                                                                       ...
                                                  </dashboard> ----------------------- </dashboard>
                                                  <seats>
                                                  ...
                                      </body> --- </body>
<gearbox> --------------------------- <gearbox>
                                      ...
                                      </car>

0                  1                  2           3                  4                 5
```

Figure 12: The best SPMA found by the SP model, with the New SP-pattern '`<crankshaft> <steering-wheel> <speed-dial> <gearbox>`' shown in column 1, and a set of Old SP-patterns representing different parts and sub-parts of a car, shown in columns 1 to 5.

The kind of structure shown in Figure 12 exhibits a form of inheritance, much like inheritance in a class-inclusion hierarchy. In this case, recognition of something as a '`<speed-dial>`' suggests that it is likely to be part of a '`<dashboard>`', which itself is likely to be part of the '`body`' of a '`<car>`'. This kind of inference is the kind of thing that crime investigators will do: search for a missing body when a severed human arm has been discovered.

C ICMUP applications

As noted in Section 2.2.3, a working hypothesis in this research is that ICMUP is fundamental not only in mathematics [41] but also in all techniques for IC, including those described in Section 2.2. This appendix tries to summarise that working hypothesis in terms of broad categories of technique.

C.1 Matching a pair of one-dimensional SP-patterns

With one-dimensional SP-patterns there are, broadly, two matching schemes.

- *Matching both SP-patterns from left-to-right.* Examples here include the techniques used in the SP research to date which are largely the techniques described in Section 2.2.

- *Mirror-matching.* Matching one SP-pattern from left-to-right and the other from right-to-left. Nothing like this has been tried with the SPCM but it should be a relatively easy thing to implement.

In both cases, the two SP-patterns would be matched, as described in [30, Appendix A] and incorporated into the SPCM, or something equivalent, which allows for zero, one, two, or more matching segments between the two SP-patterns. In other words, there can be discontinuous matches or complex partial matches, as well as exact matches between two SP-patterns.

C.2 Matching a pair of two-dimensional SP-patterns

The main possibilities with two-dimensional SP-patterns include:

- *Vertical matching.* Matching would be the same as in the two options in Section C.1, but, in addition, from top to bottom and bottom to top.

- *Create a panorama.* A panorama may be created, left to right or from right to left, from two or more partially-overlapping views.

- *The discovery of a three-dimensional digital model of an entity.* This can be done by taking several partially-overlapping views of an entity. The technique is used to create Google's Street Views, and it is also used by commercial companies to create 3D digital models of objects (see [33, Sections 6.1 and 6.2]).

- *Fusing images from two eyes, with the inference of objects and distances.* In a book published in 1971 [21], Béla Julesz presented pairs of random-dot images which, when viewed through a stereoscope would yield the perception of one or more objects suspended above a background, with or without variations in depth in the object(s). Since each of the two images was entirely random, the perception must have derived from a process of mental fusing of the two images, together with redundancies between the two images.

  What is happening here may be explained by a theory developed by Marr and Poggio [24]. Since ICMUP has an important role in that theory, we may supposed that the same is true of human perceptions of random-dot stereograms.

  Both suppositions are strengthened by the workings of a computer program developed by Grimson [17] which, in accordance with Marr and Poggio's theory, could discover the hidden image in a random-dot stereogram with performance on a late-1970s computer that "coincides well with that of human subjects" [17, Section 5].

Apart from the creation of a panorama (second item above) or creating a digital image of an object from overlapping views of the object (third item), all the matchings described above, including those in Section C.1, would have the same generality and potential for the discovery of complex partial matches, or discontinuous matches, as the matching processes in the SPCM, described in [30, Appendix A].

C.3 Established techniques for IC

The suggestion here, with only brief discussion, is that in all the established techniques for IC, such as those described by Darrel Hankerson and colleagues [18], and by Khalid Sayood [26], ICMUP, may be seen at work in either or both of two ways:

- *Via direct ICMUP.* Examples here include the popular LZ algorithms described in [26, Section 5.4].
- *Via mathematics.* Examples include arithmetic coding [26, Chapter 4] and wavelet compression [26, Chapters 15 and 16]. It seems that much of the power of methods like these derives from mathematics, which is itself founded on ICMUP [41].

In the first case, above, it is clear that ICMUP has a prominent roll.

In the second case, ICMUP may also be seen to have an important roll if it is accepted that "Mathematics [may be seen] as information compression via the matching and unification of patterns" [41] (see also Section 2.2.10).

D Mathematics incorporated in the SP Computer Model or contributing to its development

In order to demonstrate that, notwithstanding the *hundreds* of versions of the SPCM that were developed [**?**, Section 3], and the reliance on ICMUP for the modellin of inferences and unsupervised learning (Section 2.2.1), the SPCM has a fundamental rigour, illustrated in this appendix with details of mathematics that is incorporated in the SPCM or contributing to its development. It is adapted with permission from [40, Appendix A].

D.1 Searching for repeating patterns

At first sight, the process of searching for repeating patterns (Appendix **??**) is simply a matter of comparing one pattern with another to see whether they match each other or not. But there are, typically, many alternative ways in which patterns within a given body of information, **I**, may be compared—and some are better than others.

We are interested in finding those matches between patterns that represent most redundancy and thus, via unification, yield most compression—and a little reflection shows that this is not a trivial problem [30, Section 2.2.8.4].

Maximising the amount of redundancy found means maximising R where:

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

f_i is the frequency of the ith member of a set of n patterns and s is its size in bits. Patterns that are both big and frequent are best. This equation applies irrespective of whether the patterns are coherent substrings or patterns that are discontinuous within **I**.

Maximising R means searching the space of possible unifications for the set of big, frequent patterns that gives the best value. For a sequence containing N symbols, the number of possible subsequences (including single symbols and all composite patterns, both coherent and fragmented) is:

$$P = 2^N - 1. \tag{2}$$

The number of possible comparisons is the number of possible pairings of subsequences which is:

$$C = P(P - 1)/2. \tag{3}$$

For all except the very smallest values of N, the value of P is very large and the corresponding value of C is huge. In short, the abstract space of possible comparisons between patterns and thus the space of possible unifications is, in the great majority of cases, astronomically large.

Since the space is normally so large, it is not feasible to search it exhaustively. For that reason, it is normally necessary to use heuristic methods in searching—conducting the search in stages and discarding all but the best results at the end of each stage—and we must be content with answers that are “reasonably good”.

Because it is not normally possible to use exhaustive search, we cannot normally guarantee to find the theoretically ideal answer. And, normally, we cannot know whether or not we have found that theoretically ideal answer.

D.2 Information, compression of information, inductive inference and probabilities

Solomonoff [27, 28] seems to have been one of the first people to recognise the close connection that exists between IC and *inductive inference*: predicting the future from the past, and calculating probabilities for such inferences. The connection between them—which may at first sight seem obscure—lies in the redundancy-as-repetition-of-patterns view of redundancy (Appendix ??):

- Patterns that repeat within **I** represent redundancy in **I**, and IC can be achieved by reducing multiple instances of any pattern to one.

- When we make inductive predictions about the future, we do so on the basis of repeating patterns. For example, the repeating pattern 'Spring, Summer, Autumn, Winter' enables us to predict that, if it is Spring time now, Summer will follow.

Thus IC and inductive inference are closely related to concepts of frequency and probability. Here are some of the ways in which these concepts are related:

- Probability has a key role in Shannon's concept of information. In that perspective, the average quantity of information conveyed by one symbol in a sequence is:

  $$H = -\sum_{i=1}^{i=n} p_i \log p_i, \tag{4}$$

  where p_i is the probability of the ith type in the alphabet of n available alphabetic symbol types. If the base for the logarithm is 2, then the information is measured in 'bits'.

- Measures of frequency or probability are central in techniques for economical coding such as the Huffman method [11, Section 5.6] or the Shannon-Fano-Elias method [11, Section 5.9].

- In the redundancy-as-repetition-of-patterns view of redundancy and IC, the frequencies of occurrence of patterns in **I** is a main factor (with the sizes of patterns) that determines how much compression can be achieved.

- Given a body of (binary) data that has been 'fully' compressed (so that it may be regarded as random or nearly so), its absolute probability may be calculated as $p_{ABS} = 2^{-L}$, where L is the length (in bits) of the compressed data.

Probability and IC may be regarded as two sides of the same coin. That said, they provide different perspectives on a range of problems. In this research, the IC perspective—with redundancy-as-repetition-of-patterns—seems to be more fruitful than viewing the same problems through the lens of probability. In the first case, one can see relatively clearly how compression may be achieved by the primitive operation of unifying patterns whereas these ideas are obscured when the focus is on probabilities.

D.3 Random-dot stereograms

A particularly clear example of the kind of search described in Appendix **??** is what the brain has to do to enable one to see the figure in the kinds of random-dot stereogram described in [40, Section 11].

In this case, assuming the left image has the same number of pixels as the right image, the size of the search space is:

$$S = P^2/2 \tag{5}$$

where P is the number of possible patterns in each image, calculated in the same way as was described in Appendix D.1. The fact that the images are two dimensional needs no special provision because the original equations cover all combinations of atomic symbols.

For any stereogram with a realistic number of pixels, this space is very large indeed. Even with the very large processing power represented by the 10^{11} neurons in the brain, it is inconceivable that this space can be searched in a few seconds and to such good effect without the use of heuristic methods.

David Marr [23, Chapter 3] describes two algorithms that solve this problem. In line with what has just been said, both algorithms rely on constraints on the search space and both may be seen as incremental search guided by redundancy-related metrics.

D.4 Coding and the evaluation of SPMAs in terms of IC

Given an SPMA like one of the two shown in Figure 5, one can derive a *code SP-pattern* from the SPMA in the following way:

1. Scan the SPMA from left to right looking for columns that contain an SP-symbol by itself, not aligned with any other symbol.

2. Copy these SP-symbols into a code pattern in the same order that they appear in the SPMA.

The code SP-pattern derived in this way from the SPMA shown in Figure 5 is '`S PL 4 5 1 #S`'. This is, in effect, a compressed representation of those symbols in the New pattern that form hits with Old symbols in the SPMA.

Given a code SP-pattern derived in this way, we may calculate a 'compression difference' as:

$$CD = B_N - B_E \tag{6}$$

or a ‘compression ratio’ as:

$$CR = B_N / B_E, \tag{7}$$

where B_N is the total number of bits in those symbols in the New pattern that form hits with Old symbols in the SPMA, and B_E is the total number of bits in the code SP-pattern (the ‘encoding’) that has been derived from the SPMA as described above.

In each of these equations, B_N is calculated as:

$$B_N = \sum_{i=1}^{h} C_i, \tag{8}$$

where C_i is the size of the code for ith symbol in a sequence, $H_1 ... H_h$, comprising those symbols within the New pattern that form hits with Old symbols within the SPMA (Appendix D.5).

B_E is calculated as:

$$B_E = \sum_{i=1}^{s} C_i, \tag{9}$$

where C_i is the size of the code for ith symbol in the sequence of s symbols in the code pattern derived from the SPMA (Appendix D.5).

D.5 Encoding individual symbols

The simplest way to encode individual symbols in the New pattern and the set of Old patterns in an SPMA is with a ‘block’ code using a fixed number of bits for each symbol. But the SPCM uses variable-length codes for symbols, assigned in accordance with the Shannon-Fano-Elias coding scheme [11, Section 5.9] so that the shortest codes represent the most frequent alphabetic symbol types and *vice versa*. Although this scheme is slightly less efficient than the well-known Huffman scheme, it has been adopted because it avoids some anomalous results that can arise with the Huffman scheme.

For the Shannon-Fano-Elias calculation, the frequency of each alphabetic symbol type (f_{st}) is calculated as:

$$f_{st} = \sum_{i=1}^{P} (f_i \times o_i) \tag{10}$$

where f_i is the (notional) frequency of the ith pattern in the collection of Old SP-patterns (the *grammar*) used in the creation of the given SPMA, o_i

is the number of occurrences of the given symbol in the ith SP-pattern in the grammar and P is the number of SP-patterns in the grammar.

D.6 Calculation of probabilities associated with any given SPMA

As may be seen in [30, Chapter 7], the formation of SPMAs in the SP framework supports a variety of kinds of probabilistic reasoning. The core idea is that any Old symbol in a SPMA that is *not* aligned with a New symbol represents an inference that may be drawn from the SPMA. This section describes how absolute and relative probabilities for such inferences may be calculated.

D.6.1 Absolute probabilities

Any sequence of L symbols, drawn from an alphabet of $|A|$ alphabetic types, represents one point in a set of N points where N is calculated as:

$$N = |A|^L. \tag{11}$$

If we assume that the sequence is random or nearly so, which means that the N points are equi-probable or nearly so, the probability of any one point (which represents a sequence of length L) is close to:

$$p_{ABS} = |A|^{-L}. \tag{12}$$

In the SPCM, the value of $|A|$ is 2.

This equation may be used to calculate the absolute probability of the code, C, derived from the SPMA as described in Appendix D.4. p_{ABS} may also be regarded as the absolute probability of any inferences that may be drawn from the SPMA as described in [30, Section 7.2.2].

D.6.2 Relative probabilities

The absolute probabilities of SPMAs, calculated as described in the last subsection, are normally very small and not very interesting in themselves. From the standpoint of practical applications, we are normally interested in the *relative* values of probabilities, not their *absolute* values.

The procedure for calculating relative values for probabilities (p_{REL}) is as follows:

1. For the SPMA which has the highest CD (which we shall call the *reference SPMA*), identify the symbols from New which are encoded

by the SPMA. We will call these symbols the *reference set of symbols in New*.

2. Compile a *reference set of SPMAs* which includes *the SPMA with the highest CD and all other SPMAs (if any) which encode exactly the reference set of symbols from New, neither more nor less.*

3. The SPMAs in the reference set are examined to find and remove any rows which are redundant in the sense that all the symbols appearing in a given row also appear in another row in the same order.[11] Any SPMA which, after editing, matches another SPMA in the set is removed from the set.

4. Calculate the sum of the values for p_{ABS} in the reference set of SPMAs:

$$p_{A\_SUM} = \sum_{i=1}^{i=R} p_{ABS_i} \tag{13}$$

   where R is the size of the reference set of SPMAs and p_{ABS_i} is the value of p_{ABS} for the ith SPMA in the reference set.

5. For each SPMA in the reference set, calculate its relative probability as:

$$p_{REL_i} = p_{ABS_i} / p_{A\_SUM}. \tag{14}$$

The values of p_{REL} calculated as just described seem to provide an effective means of comparing the SPMAs in the reference set. Normally, this will be those SPMAs which encode the same set of symbols from New as the SPMA which has the best overall CD.

D.7 Sifting and sorting of SP-patterns in unsupervised learning in the SPS

In the process of unsupervised learning in the SPS (Appendix A.6 and [30, Chapter 9]), which starts with a set of New SP-patterns, there is a process of sifting and sorting Old SP-patterns that are created by the SPS to develop one or more alternative collections of Old SP-patterns (*grammars*), each one

---

[11] If Old is well compressed, this kind of redundancy amongst the rows of a SPMA should not appear very often.

of which scores well in terms of its capacity for the economical encoding of the given set of New SP-patterns.

When all the New SP-patterns have been processed in this way, there is a set A of full SPMAs, divided into $b_1 ... b_m$ disjoint subsets, one for each SP-pattern from the given set of New SP-patterns. From these SPMAs, the program computes the frequency of occurrence of each of the $p_1 ... p_n$ Old SP-patterns as:

$$f_i = \sum_{j=1}^{j=m} max(p_i, b_j) \tag{15}$$

where $max(p_i, b_j)$ is the maximum number of times that p_i appears in any *one* of the SPMA in the subset b_j.

The program also compiles an alphabet of the alphabetic symbol types, $s_1 ... s_r$, in the Old SP-patterns and, following the principles just described, computes the frequency of occurrence of each alphabetic symbol type as:

$$F_i = \sum_{j=1}^{j=m} max(s_i, b_j) \tag{16}$$

where $max(s_i, b_j)$ is the maximum number of times that s_i appears in any *one* SPMA in subset b_j. From these values, the encoding cost of each alphabetic symbol type is computed using the Shannon-Fano-Elias method as before [11, Section 5.9].

In the process of building alternative grammars, the tree of such alternatives is pruned periodically to keep it within reasonable bounds. Values for G, E and $(G + E)$ (which we will refer to as T) are calculated for each grammar and, at each stage, grammars with high values for T are eliminated.

For a given grammar comprising SP-patterns $p_1 ... p_g$, the value of G is calculated as:

$$G = \sum_{i=1}^{i=g} (\sum_{j=1}^{j=L_i} s_j) \tag{17}$$

where L_i is the number of symbols in the ith SP-pattern and s_j is the encoding cost of the jth SP-symbol in that SP-pattern.

Given that each grammar is derived from a set $a_1 ... a_n$ of SPMAs (one SPMA for each pattern from New), the value of E for the grammar is calculated as:

$$E = \sum_{i=1}^{i=n} e_i \tag{18}$$

where e_i is the size, in bits, of the code SP-pattern derived from the ith SPMA.

D.8 Finding good matches between two sequences of symbols

At the heart of the SPCM is a process for finding good matches between two sequences of symbols, described quite fully in [30, Appendix A]. What has been developed is a version of dynamic programming with the advantage that it can find two or more good matches between sequences, not just one good match.

The search process uses a measure of probability, p_n, as its metric. This metric provides a means of guiding the search which is effective in practice and appears to have a sound theoretical basis. To define p_n and to justify it theoretically, it is necessary first to define the terms and variables on which it is based:

- A sequence of matches between two sequences, sequence1 and sequence2, is called a 'hit sequence'.
- For each hit sequence $h_1 ... h_n$, there is a corresponding series of *gaps*, $g_1 ... g_n$. For any one hit, the corresponding gap is $g = g_q + g_d$, where g_q is the number of unmatched characters in the query between the query character for the given hit in the series and the query character for the immediately preceding hit; and g_d is the equivalent gap in the database, g_1 is taken to be 0.
- A is the size of the *alphabet* of symbol types used in sequence1 and sequence2.
- p_1 is the probability of a match between any one symbol in sequence1 and any one symbol in sequence2 on the null hypothesis that all hits are equally probable at all locations. Its value is calculated as: $p_1 = 1/A$.

Using these definitions, the probability of any hit sequence of length n is calculated as:

$$p_n = \prod_{i=1}^{i=n} (1 - (1 - p_1)^{g_i + 1}), \quad g_1 = 0$$

.

With this equation, is relatively easy to calculate the probability of the hit sequence up to and including any hit by using the stored value of the hit sequence up to and including the immediately preceding hit.

E Abbreviations

The abbreviations used in this paper are shown here in alphabetical order:

- *Artificial intelligence*: 'AI'.
- *Human learning, perception, and cognition*: 'HLPC'.
- *Information compression*: 'IC'.
- *Information compression via the matching and unification of patterns*: 'ICMUP'.
- *New Mathematics*: 'NM'.
- *Quantum Mechanics*: 'QM'.
- *SP-multiple-alignment*: 'SPMA'.
- *Standardised form for knowledge*: 'STFK'.
- *Universal Framework for the representation and processing of diverse kinds of Knowledge*: 'UFK'.

The name 'SP' originates in two features of the research: 1) The SP Theory is intended, in itself, to combine *Simplicity* with descriptive and explanatory *Power*; and 2) The SPS works entirely by the compression of a given body of information, **I**, and this may be seen as a process that promotes *Simplicity* in **I** whilst retaining as much as possible of the descriptive and explanatory *Power* of **I**.

However, *it is intended that 'SP' should be treated as a name, without any need to expand the letters in the name or explain the origin of the letters, as with names such as 'IBM' or 'BBC'.*

References

[1] J. Al-Khalili. *Quantum: A Guide for the Perplexed.* Phoenix, London, Kindle edition, 2012.

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

[3] F. Attneave. *Applications of Information Theory to Psychology.* Holt, Rinehart and Winston, New York, 1959.

[4] P. Ball. *Beyond Weird.* Vintage, London, Kindle edition, 2018.

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

[8] M. Born. *Born-Einstein Letters.* Walker Publishing, New York, second edition, 2005. Originally published in 1971, with new material for the 2005 edition.

[9] J. Brockman. *This Idea Must Die.* HarperCollins, London, Kindle edition, 2015.

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

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

[12] P. A. M. Dirac. *The Principles of Quantum Mechanics.* Snowball Publishing, Northern Illinois, fourth, Kindle edition, 1947.

[13] M. Dyakonov. The case against quantum computing. *IEEE Spectrum*, 15 Nov 2018, 2018.

[14] B-G. Englert. On quantum theory. *arXiv*, page 16, 2013. arXiv:1308.5290. Later published in Englert, B. On quantum theory. *The European Physical Journal D* volume 67, 238 (2013). https://doi.org/10.1140/epjd/e2013-40486-5.

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

[16] S. Greengard. The algorithm that changed quantum machine learning. *Communications of the ACM*, 62(8):15–17, 2019.

[17] W. E. L. Grimson. A computer implementation of a theory of human stereo vision. *Proceedings of the Royal Society of London. Series B*, 292(1058), 1981. A draft copy may be downloaded via bit.ly/1Jq8zYk.

[18] D. Hankerson, G. A. Harris, and P. D. Johnson Jr. *Introduction to Information Theory and Data Compression.* Chapman & Hall/CRC, London, second (Kindle) edition, 2003.

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

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

[21] B. Julesz. *Foundations of Cyclopean Perception.* Chicago University Press, Chicago, 1971.

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

[23] D. Marr. *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, San Francisco, 1982.

[24] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B*, 204(1156):301–328, 1979.

[25] V. Palade and J. G. Wolff. A roadmap for the development of the 'SP Machine' for artificial intelligence. *The Computer Journal*, 62:1584–1604, 2019. https://doi.org/10.1093/comjnl/bxy126, arXiv:1707.00614, bit.ly/2tWb88M.

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

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

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

[29] J. G. Wolff. Medical diagnosis as pattern recognition in a framework of information compression by multiple alignment, unification and search. *Decision Support Systems*, 42:608–625, 2006. arXiv:1409.8053 [cs.AI], bit.ly/1F366o7.

[30] 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. The print version is produced via print-on-demand from "INGRAM Lightning Source" and, via that technology, will never go out of print.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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, Boca Raton, Florida, 2016, Chapter 6, pp. 143–170.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] J. G. Wolff. Mathematics as information compression via the matching and unification of patterns. *Complexity*, 2019:25, 2019. Article ID 6427493, Archives: vixra.org/abs/1912.0100 and hal.archives-ouvertes.fr/hal-02395680. This paper is reproduced in *New Ideas Concerning Science and Technology*, Vol. 13, 12 June 2021, Pages 132–169, doi.org/10.9734/bpi/nicst/v13/8449D.

[42] J. G. Wolff. Biochemistry provides inspiration for a new kind of ai. 2021. tinyurl.com/473dawp8, Submitted for publication.