Intelligence Via Compression of Information

A Tutorial

The SP-Theory of Intelligence*, and its realization in the* SP Computer Model*, can demonstrate diverse aspects of intelligence, all achieved by compressing information via the powerful* SP-Multiple-Alignment *construct. Quite unexpectedly, these ideas have led to an important new insight in the foundations of mathematics, and to new thinking about the concepts of probability and computing.*

J Gerard Wolff*

Thursday 14$^{\text{th}}$ March, 2024, 12:20

Abstract

As the title of this book suggests, it is about how intelligence may be understood as information compression (IC). More specifically, the book is about the *SP Theory of Intelligenc* (SPTI) and its realisation in the *SP Computer Model*—and their potential applications, benefits, and associated ideas. The SPTI draws on substantial evidence for the importance of IC in human learning, perception, and cognition. Since the SPTI also has much to say about issues in artificial intelligence (AI), it is a theory of both natural and artificial intelligence. In the SPTI, IC is achieved largely via the powerful concept of *SP-Multiple-Alignment*, a major discovery which is largely responsible for the versatility of the SPTI in aspects of human intelligence and beyond. Strengths of the SPTI include: the modelling of several kinds of intelligent behaviour, including several kinds of probabilistic reasoning; the representation and processing of several kinds of intelligence-related knowledge; and the seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination. That seamless integration appears to be *essential* in any AI system that aspires to the fluidity and versatility of human-level intelligence. Related to the SPTI is another major discovery: *that mathematics may be seen as a set of techniques for IC, and their application.* This suggests the creation of a *New Mathematics* via the integration of mathematics with the SPTI, combining the strengths of both. The SPTI also suggests new thinking in concepts of probability and new thinking about 'computation, with potential benefits in both areas. The SPTI has been shown in peer-reviewed papers to be relevant to areas not closely associated with AI. These include: the management of 'big data'; the development of autonomous robots; medical databases; sustainability of computing; transparency in computing; and computer vision.

Keywords: SP Theory of Intelligence; SP Computer Model; information compression; artificial intelligence; SP-Multiple-Alignment, processing of natural language, pattern recognition, planning, problem solving.

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

Contents

1 Introduction

As the title of this book suggests, it is about intelligence and how it may be understood as compression of information. More specifically, it is about the *SP-Theory of Intelligence* (SPTI), its realisation in the *SP Computer Model* (SPCM), and their potential benefits and applications. In general, references to the SPTI should be understood to include the SPCM unless stated otherwise.

This book is largely a tutorial about ideas previously described in [1] together with ideas from later research. It is free-standing with new thinking and new perspectives, and without the need to refer to the earlier book.

Since the development of the SPTI draws on evidence relating to intelligence in human learning, perception, and cognition, and since the SPTI has things to say about issues in AI, the SPTI may be seen to be a theory of both natural and artificial intelligence. As we shall see, the SPTI also has things to say about neuroscience (Section 9), and somewhat unexpectedly, the foundations of mathematics (Section 10).

Although the SPTI, like most theories, is not a comprehensive theory of what it aims to describe—intelligence in this case—the SPTI should best be regarded as a *foundation* for the development of human-level intelligence, aka 'artificial general intelligence' (AGI), and that further development will be needed, as outlined in [2], before the SPTI can be described as a theory of AGI (Section 8.2.2).

1.1 My background and experience

As a possible aid to understanding how these ideas developed (Section 5), here is a brief summary of my background and experience.

I studied natural sciences at Cambridge University (zoology, botany, geology, specialising in psychology), followed by two years' school teaching as a requirement for a one-year course in educational psychology. This led to research into the learning of a child's first language with the building of computer models, first in an MSc, and later in a PhD, both in the University of Wales, Cardiff.

For several years I was a lecturer in psychology at the University of Dundee, followed by a one-year fellowship with IBM in Winchester (helping to develop a system for translating speech into text), followed by several years work with a software company in the city of Bath. After gaining valuable experience there, I returned to academic work, with several years' teaching and research in AI and software engineering in the School of Electrical and Electronic Engineering in the University of Wales, Bangor (which later became the School of School of Computer Science and Electronic Engineering in Bangor University).

My research developing the SPTI and the SPCM was begun and taken to an advanced stage with my job in Bangor and has continued since I opted for an early retirement scheme established by the university.

1.2 The main ideas in this book

The SPTI, and some of its potential benefits and application are described most fully in [1] and more briefly in [3]. Other documents, including peer-reviewed papers, are detailed with download links on www.cognitionresearch.org/sp.htm.

This book covers the following main ideas:

- *The importance of IC across diverse aspects of natural intelligence.* A foundation for much of this research is substantial evidence, mentioned above, for the importance of IC across diverse aspects of intelligence in people and other animals (Section 2). In keeping with that evidence, IC is fundamental in how the SPTI models diverse aspects of intelligence.

  This contrasts with other approaches to the development of AI where the relevance of IC is recognised largely in unsupervised learning [4, 5], and then only to a limited extent,

  It appears that the SPTI is the only theory in which IC is fundamental in all the several aspects of intelligence modelled by the theory, with the expectation that IC will be fundamental in any other aspect of intelligence, if any, not yet addressed by the theory.

- *The SP-Multiple-Alignment concept* (Section 6.3). The SP-Multiple-Alignment concept is largely responsible for the versatility of the SPTI in modelling diverse aspects of human intelligence (Sections 8.1 and 8.2), and in other areas less closely-related to AI (Section 8.3).

Although the SP-Multiple-Alignment concept is far from being trivially simple, it is remarkably simple in view of the versatility that it imparts to the SPTI. In short, the SP-Multiple-Alignment concept is largely responsible for the SPTI's favourable combination of Simplicity with Power (Appendix B).

It is no exaggeration to say that **the SP-Multiple-Alignment concept is a *major discovery* with the potential to be as significant for an understanding of intelligence as is the concept of DNA for an understanding of biology. It may prove to be the double helix of intelligence!** (Section 6.3.1).

- *Examples of the versatility of the SP-Multiple-Alignment concept within the spcm* (section 7). the examples in this section demonstrate much of the versatility of the SP-Multiple-Alignment framework for modelling diverse aspects of intelligence.

- *Notable strengths of the SPTI.* Apart from the SP-Multiple-Alignment and its versatility, strengths of the SPTI that deserve special mention include:

  - *The need for transparency in the organisation and workings of the SPTI* ([6, Section 10]). Unlike deep neural networks (DNNs), the SPCM provides an audit trail for all its workings, and there is transparency in the way it structures knowledge. These features are likely to prove useful in legal disputes and in minimising potential risks from AI.
  - *Much reduced demands for data and computational resources compared with DNNs* ([7, Sections VII, VIII, and IX], [6, Section 9]). There is clear potential for big reductions in the huge demands for data and for computational resources by DNNs, and technologies that exploit DNNs such as 'large language models' and 'generative AIs'.
  - *Generalisation, over-generalisation, and under-generalisation* (Section 6.7.4, [6, Section 6]). The SPTI framework of ideas suggests that remarkably simple principles govern the phenomena of generalisation, and the correction of over- and under-generalisations.
  - *In learning, reducing or eliminating the corrupting effect of errors in data* (Section 6.7.5.
  - *How to learn usable knowledge from a single exposure or experience* ([6, Section 7]). Like people and unlike DNNs, the SPTI can learn usable knowledge from a single exposure or experience.

- *Mathematics as information compression* (Section 10). A second *major discovery* is that **mathematics may be seen as a set of techniques for IC, and their application** [8]. This has a bearing on related issues:

  - *The foundations of mathematics.* This view of mathematics is a radical alternative to existing isms in the foundations of mathematics [8, Section 2].
  - *Why is mathematics so effective in science?.* The IC view of mathematics provides an answer to the often-repeated question: 'Why is mathematics so effective in science?' (Section 10.7).
  - *A New Mathematics.* The idea that IC is central in both the SPTI and mathematics suggests the creation of a *New Mathematics* as an integration of the two, with potential benefits that appear to be substantial (Section 11).

  Logic and computing may also be understood in terms of IC [8, Section 7].

- *IC as the basis for both the compression and decompression of knowledge.* Paradoxical as it may sound, IC is the basis within the SPTI for both the compression and decompression of knowledge (Section 8.3.3).

- *Seamless integration of diverse aspects of intelligence, in any combination.* An important strength of the SPTI is that it provides for the seamless integration of diverse aspects of intelligence in any combination (Section 8.1.4).

  This strength, which arises from the provision of a single framework for diverse aspects of intelligence and diverse kinds of intelligence-related knowledge, appears to be *essential* in any theory of AI that aspires to model the fluidity and versatility of human intelligence.

- *The SPTI as a relatively firm foundation for the development of human-level intelligence.* Notwithstanding impressive results obtained with the currently-popular DNNs, the SPTI provides a firmer foundation for the development of AGI, than DNNs [6, 9].

- *SP-Neural.* SP-Neural is a preliminary version of the SPTI which expresses abstract concepts in the SPTI in terms of neurons, connections between neurons, and their inter-communications ([10], [1, Chapter 11]). It seems likely that inhibition, which is a prominent feature of neural tissue, lies at the heart of how brains and nervous systems achieve IC (Section 9.1).

  SP-Neural as it develops has a better chance than DNNs of reflecting the organisation and workings of real neural networks (Section 9.2).

- *Potential of the SPTI as a theory of probabilities.* Apart from its strengths as a theory of human learning, peception, and cognition and AI, the SPTI has potential as a theory of probabilities, with substantial potential benefits (Section 12).

- *Potential of the SPTI as a theory of computing.* Apart from its strengths as a theory of human learning, peception, and cognition and AI, the SPTI has potential as a theory of computing, with substantial potential benefits (Section 13).

1.3 Alternative perspectives

Readers familiar with the kinds of issues discussed in this book, may wonder why there is relatively little said about Algorithmic Information Theory (AIT), Bayes' theorem, DNNs, or why mathematics is not more prominent in the book (although it is used in many parts of the SPCM as may be seen in [11, Appendix A]).

This section outlines why these things have not featured large in the research that is the subject of this book.

1.3.1 Algorithmic information theory

The IC theme in the SPTI may be seen as an example of Ockham's razor and is also central in a body of inter-related ideas associated with such names as the afore-mentioned AIT, 'Algorithmic Probability Theory', 'Minimum Description Length', 'Minimum Message Length', and 'Kolmogorov Complexity' [12]. In the rest of this book, all these inter-related ideas are referred to as 'AIT', except in Appendix D where the focus is on Algorithmic Probability Theory (APT).

A key idea in AIT is that the non-redundant information content of a body of information, **I**, is the length of the smallest computer program that outputs **I**, where 'smallest' means the smallest that it has been possible to achieve via IC with the available computational resources.

Another important idea, from Solomonoff‘s APT (Appendix D), is that compression of information is closely related to concepts of probability.

The SPTI is a computational version of Ockham's razor, and as such it is broadly consistent with AIT. But otherwise the two fields are radically different.

In comparison with AIT and related ideas, distinctive features of the SPTI are:

- That a key part of the SPTI is the SP-Multiple-Alignment concept (Section 6.3) which provides an effective means of compressing diverse kinds of information.
- Thanks largely to the SP-Multiple-Alignment concept, the SPTI has strengths in the modelling of diverse aspects of intelligence including several forms of probabilistic reasoning, the representing and processing of diverse kinds of intelligence-related knowledge, and providing the basis for the seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination (Section 8).
- That, while AIT uses a measure of IC based on the universal Turing machine, the SPTI uses a concept of IC based on the primitive operation of merging or 'unifying' patterns that are the same, or parts thereof, within the SP-Multiple-Alignment framework within the SPTI (Section 4).
- That the importance of IC in the SPTI suggest new foundations for mathematics which are radically different from any of the existing 'isms' in the foundations of mathematics (Section 10).
- There is potential for the SPTI to be integrated with mathematics, thus creating a *New Mathematics* with many potential advantages (Section 11);
- The SPTI has potential as the basis for new concepts in probability (Section 12);
- And the SPTI has potential as the basis for a new model of computing (Section 13).
- By contrast with AIT, in which there is research relating AIT to Bayes' theorem (see, for example, [12, pp. 350–353] and APT [13, p. 75 and p. 78]), the SPTI is radically different from Bayes' theorem with little prospect of any kind of rapprochement (Section 1.3.2).

1.3.2 Bayes' theorem

Bayes' theorem is defined mathematically as follows:

$$P(A|B) = (P(B|A)P(A))/(P(B)) \tag{1}$$

where A and B are 'events' and $P(B) \neq 0$.[1] Within the equation:

- $P(A|B)$ is the conditional probability of event A occurring given that B is true. It is also called the posterior probability of A given B.

[1] Here, an 'event' is a set of outcomes of an experiment to which a probability is assigned.

- $P(B|A)$ is the conditional probability of event B occurring given that A is true. It can also be interpreted as the likelihood of A given a fixed B because $P(B|A) = L(A|B)$.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B respectively without any given conditions; they are known as the prior probability and marginal probability.

Bayes' Theorem Would Be More of a Hindrance Than a Help in Developing the SPTI Although Bayes' theorem has inspired much research, it would probably be more of a hindrance than a help in the development of the SPTI. The main reasons are these:

- Bayes' theorem assumes that entities or 'events' to which the theorem applies are already in existence. There is no place in the theorem for the unsupervised learning of new entities and new structures, as described in Section 2.4 and Section 6.7.6.
- Concepts of 'posterior probability', 'prior probability', and 'marginal probability' in Bayes' theorem do not fit into the SPTI framework. Despite criticisms of 'frequentist' theories of probability, the SPTI has incorporated the simple idea that probabilities may be derived straightforwardly from frequencies. The frequency of any SP-Pattern is the frequency of the SP-Pattern is the number of SP-Patterns from which, via unifications, it is derived—unless it has not been created via unification, in which case the SP-Pattern's frequency is 1.
- Despite the intimate relation between IC and concepts of probability (Appendix D), and despite abundant evidence for the importance of IC in human learning, peception, and cognition (Section 2), IC has no place in Bayes' theorem.

Modelling an application of Bayes' Theorem with the SPCM Although Bayes' theorem has no place in the SPTI, Bayesian networks can be modelled in the SPCM, as described in [1, Section 7.8].

1.3.3 Deep neural networks

The programme of research to develop the SPTI was started before DNNs became popular, but, when DNNs became prominent, there was no temptation to adopt the DNN technology mainly because of the shortcomings listed below.

With respect to the SPTI as it is now, and likely to develop in the future, DNNs have several shortcomings:

1. *No Firm Commitment to IC Across All Aspects of AI.* In the development of DNNs there is some acknowledgement here and there that IC might have a place in machine learning (see, for example, [4, Sections 4.4, 5.6.3 and 6.7]) but there is no recognition of the central importance of IC across all aspects of intelligence in humans and other mammals (Section 2) and what that means for AI.

   That uncertainty persists in a recent paper [5] with the focus on DNNs, with a title that begins 'To Compress or Not to Compress ...', and again with the focus on machine learning, and without any recognition that IC might be relevant to other aspects of intelligence.

2. *Too Much Reliance on an Implausible Neural Architecture.* DNNs evolved from 'artificial neural networks' which themselves were widely understood to be approximations of real neural networks. With the introduction of such things as backpropagation (with little relation to neural backpropagation) and the multiple layers of DNNs, ideas about structures and processes have digressed more and more from biological underpinnings.

   As noted in Section 1.2, last bullet point but two, SP-Neural as it develops has a better chance than DNNs of reflecting the organisation and workings of real neural networks (Section 9.2).

3. The huge demands of DNNs for data.

4. The huge demands for computation with corresponding demands for energy, contrasting sharply with approximately 20 W needed by the human brain [14].

5. *Other Shortcomings of DNNs.* As its title suggests, the paper 'Twenty significant problems in AI research, with potential solutions via the SP Theory of Intelligence and its realisation in the SP Computer Model' [6] describes those twenty problems and how they may be solved via the SPTI/SPCM.

   Since most of those problems are problems with DNNs, the paper is in effect a critique of the DNN concept, together with putative solutions via the SP system.

   The main points in the paper are summarised in Section 8.2.1, together with a further point relating specifically to DNNs.

1.3.4 How mathematics may sometimes be a hindrance in science

This section describes some of the reasons why mathematics is less prominent in this research than in some other areas of computer science.

Despite more than 2000 years of experience with mathematics, and despite extraordinary successes with mathematics, especially in physics, it seems that mathematics can sometimes be more of a hindrance than a help in the development of scientific theories:

- It seems that leading scientists and others often adopt a 'visual' kind of thinking that is not easily expressed in mathematics, despite the existence of branches of mathematics such as geometry and topology. For example, Carlo Rovelli writes that 'Einstein had a unique capacity to imagine how the world might be constructed, to "see it in his mind." [15, location 1025].

- It seems that mathematics is geared mainly to the discovery or invention of neat equations such as Pythagoras's theorem ($h^2 = a^2 + b^2$) or Boyle's law ($PV = k$), and is less well adapted to the expression of more complex concepts such as heuristic search (Appendix G), the SP-Multiple-Alignment concept (Section 6.3), and unsupervised learning (Section 6.7).

- In addition, there is a long tradition that mathematics is to be interpreted and evaluated by human brains. It is only relatively recently that computers have been harnessed to assist with those tasks, but old traditions linger on.

For these kinds of reasons, writing programs and running them on computers can be more useful than trying to express everything with unassisted brain power or with mathematics—but mathematics may be included in those programs where necessary.

In general the advantages of using programs with computers (rather than unassisted brain power) to assist in the development of theory include:

- Providing much more flexibility than mathematics for the expression of such things as heuristic search and other things mentioned above.
- Unlike people, computers don't get tired and make mistakes.
- Computers can calculate much faster than people.
- Computers can enforce rigour in defining concepts and theories.
- And they provide a means of testing ideas, and demonstrating how a theory works.

For those kinds of reasons I have found the object-oriented computer language C++ a better medium for both the expression and testing of theoretical ideas (Section 5.4). That said, mathematics is used in several parts of the SPTI, as summarised in [11, Appendix A]. As a derivative of the C programming language, C++ has the advantage that, where necessary, one can control details of the underlying machine.

It is pertinent to mention here that the development of object-oriented computer languages, beginning with the simulation language Simula [16] amd now including many mainstream computer languages including C++, was driven mainly by the realisation that software was much easier to develop, to understand, and to maintain, when it is designed to represent real-world objects and their interactions in any given area of application. Those advantages apply just as much to software for scientific research as to software for commerce, industry, or administration.

1.4 Presentation

The sections and subsections in this book, with live links, are detailed in the Table of Contents.

As indicated by the reference to live links, the book is designed to be read online, so that readers may have the convenience of live links, and so that computer-powered searching largely eliminates the need for an index.

Details of the main papers in this programme of research, with download links, may be seen via tinyurl.com/mpwdusrx. There is more detail in www.cognitionresearch.org/sp.htm.

Somce key terms are defined in Appendix A.

There is a note on copyright in Appendix H.

2 Information compression in human learning, perception, and cognition

As noted in Section 1.2, a foundation for the SPTI and related research is substantial evidence for the importance of IC in human learning, peception, and cognition. This main section describes some of this evidence, drawing on [11], where further information may be found.

2.1 Pioneering research by Fred Attneave and Horace Barlow and others

As noted in Section 1.2, the idea that much of the workings of brains and nervous systems may be understood as IC was pioneered by Fred Attneave [17, 18], Horace Barlow [**?**, 19], and others.

This idea has been investigated by various researchers up to the present (e.g. [20, 21, 22]), much of it within the framework of AIT (Section 1.3.1).

2.1.1 Some informational aspects of visual perception

In a paper called 'Some informational aspects of visual perception', Fred Attneave [17] describes evidence that visual perception may be understood in terms of the distinction between areas in a visual image where there is much redundancy, and boundaries between those areas where non-redundant information is concentrated.:

> ... information is concentrated along contours (i.e., regions where color changes abruptly), and is further concentrated at those points on a contour at which its direction changes most rapidly (i.e., at angles or peaks of curvature).

For those reasons, he suggests that:

> Common objects may be represented with great economy, and fairly striking fidelity, by copying the points at which their contours change direction maximally, and then connecting these points appropriately with a straight edge. [17, p. 185].

And he illustrates the point with a drawing of a sleeping cat reproduced in Figure 1.

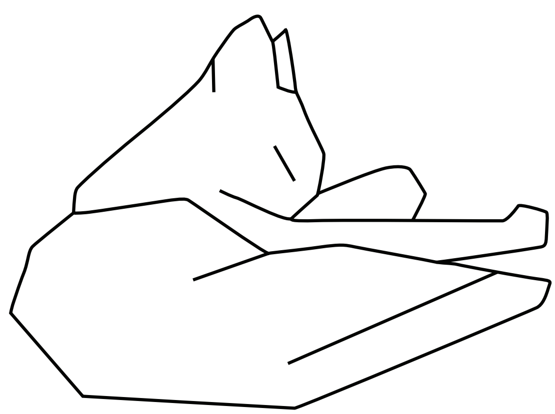

Figure 1: Drawing made by abstracting 38 points of maximum curvature from the contours of a sleeping cat, and connecting these points appropriately with a straight edge. Reproduced from Figure 3 in [17], with permission.

2.1.2 The need for compression of sensory information

In [?], Barlow argues, on the strength of the large amounts of sensory information being fed, via sense organs, into the central nervous system, and evidence that, in mammals at least, each optic nerve is too small, by a wide margin, to carry reasonable amounts of the visual information impinging on the retina, that 'the storage and utilization of this enormous sensory inflow would be made easier if the redundancy of the incoming messages was reduced.' [?, p. 537].[2]

2.1.3 IC and intelligence

In connection with the quest to understand human intelligence, it is of interest that, more than 60 years ago, a possible connection between IC and intelligence was recognised by Barlow in [?]. Later he wrote:

> ... the operations needed to find a less redundant code have a rather fascinating similarity to the task of answering an intelligence test, finding an appropriate scientific concept, or other exercises in the use of inductive reasoning. Thus, redundancy reduction may lead one towards understanding something about the organization of memory and intelligence, as well as pattern recognition and discrimination. [19, p. 210],

where 'find[ing] a less redundant code' leads to 'redundancy reduction' which means lossless IC.

In short, Barlow put his finger on the principle that, to a large extent, human intelligence means IC. This is a principle for which there is now abundant evidence, much of it described in Section 2. There is further evidence in the way that the SPTI can, via IC, reproduce several different aspects of human intelligence, without any additional learning or programming (Sections 8.1 and 8.2).

2.1.4 Barlow's change of view

In several papers, over a period of years, Barlow developed the ideas outlined in the previous subsections. However, he later adopted a new position, arguing that:

> ... the [compression] idea was right in drawing attention to the importance of redundancy in sensory messages because this can often lead to crucially important knowledge of the environment, but it was wrong in emphasizing the main technical use for redundancy, which is compressive coding. The idea points to the enormous importance of estimating probabilities for almost everything the brain does, from determining what is redundant to fuelling Bayesian calculations of near optimal courses of action in a complicated world. [23, p. 242].

While there are some valid points in what Barlow says in support of his new position, his overall conclusions appear to be wrong. His main arguments are summarised in [11, Appendix B], with what I'm sorry to say are my critical comments after each one. I feel apologetic about this because Barlow's lectures and his earlier research on the role of IC in brains and nervous systems have been an inspiration for me over many years.

Notwithstanding Barlow's change of view, the following subsections describe further evidence for the importance of IC in human learning, peception, and cognition.

[2]Here, 'redundancy' in the title of Barlow's paper means 'repetition of information', so 'reduction of redundancy' means lossless compression of information.

2.2 Storage and transmission of information

In an abstract biological view, IC can confer an advantage via natural selection to any creature:

- By allowing it to store more [not compressed] information in a given storage space;
- Or to use less storage space for a given amount of [not compressed] information.

Likewise, IC can be useful:

- By speeding up the transmission of any given volume of [not compressed] information along nerve fibres, thus speeding up reactions by a person or other animal;
- Or by reducing the bandwidth needed for the transmission of the same volume of [not compressed] information in a given time.

2.3 IC and concepts of probability

Perhaps more important than the impact of IC on the storage or transmission of information is the close connection between IC and concepts of probability (Appendix D). Compression of information provides a means of predicting the future from the past and estimating probabilities so that, for example, an animal may learn to predict where food may be found, or where there may be dangers, and so on.

This makes sense in terms of 'information compression via the matching and unification of patterns' (ICMUP, Section 4): any repeating pattern can be a basis for inferences, and the probabilities of such inferences may be derived from the number of repetitions of the given pattern.

For any animal, including the human animal, being able to estimate probabilities can mean large savings in the use of energy and other benefits in terms of survival.

There is more about IC and concepts of probability in Section 12.

The way in which the SPTI calculates absolute and relative probabilities for each SP-Multiple-Alignment is described in Section 6.6.

2.4 Research on the learning of a first language

More by luck than judgement with respect to the importance of IC within human learning, peception, and cognition and AI, I began research creating computer models of aspects of the learning of a first language by young children.[3] Both phases of that research, described in [24], and outlined in the next two subsections, yield strong evidence for the importance of IC in those two aspects of learning.

2.4.1 The segmentation problem in the learning of a first language

Early models aimed to explore how a young child, before learning to read, and without any 'teaching' or anything equivalent, can get to understand that speech is composed of segments such as words although ordinary speech has no systematic markers of the beginnings and ends of words. Computer models were created to see whether it was possible to discover word segments by heuristic search

[3]this section is based largely on [24].

(see Appendix G) with text in which all spaces and punctuation had been removed, and starting with a 'dictionary' of candidate words containing only the 26 letters of the alphabet.

The first measure of success that was used for the creation of new segments for inclusion in the dictionary was the left-to-right transition probability between two neighbouring segments from which a new segment might be created. This gave results that were reasonably encouraging but far from perfect. The same was true of a measure derived from nonparametric statistics.

The discovery of words in unsegmented Text For words or word-like segments, the best results by far were obtained from the selection of high-frequency pairs of neighbouring elements. That measure translates fairly directly into a measure of the IC that may be achieved via chunking-with-codes: treating high-frequency pairs as relatively large chunks to be replaced in the text by relatively small codes.

This discovery of word structure by the MK10 program, illustrated in Figure 2, is achieved without the aid of any kind of externally-supplied dictionary or other information about the structure of English. The program builds its own dictionary via 'unsupervised' learning using only the unsegmented sample of English with which it is supplied. It learns without the assistance of any kind of 'teacher', or data that is marked as 'wrong', or the grading of samples from simple to complex (*cf.* [25]).

Figure 2: Part of a parsing created by the MK10 Computer Model from a 10,000 letter sample of English (book 8A of the Ladybird Reading Series) with all spaces and punctuation removed. The program derived this parsing from the sample alone, without any prior dictionary or other knowledge of the structure of English. Reproduced from Figure 7.3 in [24], with permission.

Statistical tests show that the correspondence between the computer-assigned word structure and the original (human) division into words is significantly better than chance.

The discovery of phrases in unsegmented text With some adaptation of the problem—replacing each word segment with a symbol representing the grammatical category of the given word[4]—statistically significant results for the discovery of phrase structure in unsegmented text were obtained with the MK10 program [26].

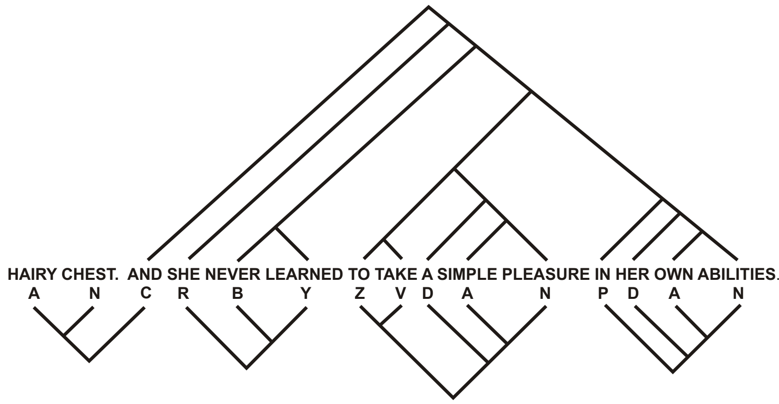

Figure 3: Phrase structure for one sentence processed as described in the text. The 'correct' parsing, via human judgement, is shown above the sentence, and the best parsing by the MK10 program is shown beneath the sentence. Reproduced from Figure 1 (sentence 10) in [26], with permission.

In short, this research—in the discovery of both words and phrases in unsegmented text—confirms the importance of IC in the analysis needed to discover segmental structures in natural language.

2.4.2 Discovering the grammars of english-like artificial languages

In later research, computer models were designed for the 'unsupervised' learning of English-like artificial languages via IC, where 'unsupervised' means that there is nothing like a 'teacher' giving positive or negative reinforcements, and there is no other aid to learning, except measures of IC as a guide to heuristic search (Appendix G).

As with the segmentation problem (Section 2.4.1), heuristic search, just mentioned, is normally needed in the unsupervised learning of SP-grammars. And, as with the segmentation problem, the best results by far were obtained when the criterion for selection at the end of each of several stages of processing was the amount of compression that had been achieved, thus strengthening the evidence for the importance of IC in human learning, peception, and cognition.

[4]I am very grateful to Dr. Isabel Forbes (with specialist knowledge of theoretical linguistics) for undertaking this task, and also for providing 'correct' parsings of the text for comparison with the computer-generated parsing, an example of which is shown in Figure 3.

3 In the quest for artificial general intelligence, the benefits of a top-down, breadth-first research strategy with wide scope

As its title suggests, this section is about how, in the quest for AGI, we may benefit from the adoption of a top-down, breadth-first research strategy with wide scope.

The alternative strategy, which is almost universal in AI research, is bottom-up, depth-first research, concentrating one's efforts on one small aspect of intelligence, with the implicit expectation that it will be possible to generalise via many such studies, and thus reach AGI.

Some of the problems associated with bottom-up, depth-first research are described in subsections that follow.

3.1 The problem of fragmentation in the development of theory

Various authors have drawn attention to the problem of fragmentation and related issues in research, with implications for the ways in which research is or should be done.

3.1.1 Allen Newell and 'You can't play 20 questions with nature and win'

Allen Newell was one of the first people to draw attention to the problems of fragmentation in cognitive science in his famous paper 'You can't play 20 questions with nature and win' [27]. In that paper, he exhorted researchers to tackle 'a genuine slab of human behaviour' [27, p. 303], thus avoiding the weaknesses of micro-theories with limited scope for generalisation. In effect, Newell was urging researchers to adopt a top-down, breadth-first strategy,

This thinking led to his book *Unified Theories of Cognition* [28] and a programme of research developing the Soar cognitive architecture [28, 29], aiming for a unified theory of cognition.

3.1.2 Pamela McCorduck and fragmentation of research

Again, in connection with fragmentation in AI, science writer Pamela McCorduck writes [30, p. 417]:

> The goals once articulated with debonair intellectual verve by AI pioneers appeared unreachable ... Subfields broke off—vision, robotics, natural language processing, machine learning, decision theory—to pursue singular goals in solitary splendor, without reference to other kinds of intelligent behaviour.

Later, she writes of 'the rough shattering of AI into subfields ... and these with their own sub-subfields—that would hardly have anything to say to each other for years to come.' [30, p. 424]. She adds: 'Worse, for a variety of reasons, not all of them scientific, each subfield soon began settling for smaller, more modest, and measurable advances, while the grand vision held by AI's founding fathers, a general machine intelligence, seemed to contract into a negligible, probably impossible dream.' [30, p. 424].

Although this was published in 2004, what McCorduck says is still true today. To a large extent, different aspects of AI are still developed independently of each other. Even when the overall goal is to develop AGI, it is commonly assumed that this may be approached via one or other particular aspects of AI. It appears that this 'bottom-up' strategy always fails, as described next.

3.1.3 Fragmentation of research at IBM

Writing about fragmentation in industrial research, John Kelly and Steve Hamm (both of IBM) write:

> Today, as scientists labor to create machine technologies to augment our senses, there's a strong tendency to view each sensory field in isolation as specialists focus only on a single sensory capability. Experts in each sense don't read journals devoted to the others senses, and they don't attend one another's conferences. Even within IBM, our specialists in different sensing technologies don't interact much. [31, location 1004].

3.2 The seductive plausibility of bottom-up research strategies, and why they fail

The reason that bottom-up research strategies are so attractive for researchers in AI seems to be that it allows researchers to concentrate on one aspect of intelligence at any one time. A bottom-up research strategy also means that new scientific papers can be produced quickly, thus helping researchers to cope with the over-strong requirement that they should 'publish or perish'.

In that connection but with reference to research in psychology, Newell writes:

> Every time we find a new phenomenon—every time we find PI release, or marking, or linear search, or what-not—we produce a flurry of experiments to investigate it. We explore what it is a function of, and the combinational variations flow from our experimental laboratories. ... in general there are many more. Those phenomena form a veritable horn of plenty for our experimental life—the spiral of the horn itself growing all the while it pours forth the requirements for secondary experiments. ... Suppose that in the next thirty years we continued as we are now going. Another hundred phenomena, give or take a few dozen, will have been discovered and explored. ... Will psychology then have come of age? Will it provide the kind of encompassing of its subject matter—the behavior of man—that we all posit as a characteristic of a mature science? ... it seems to me that clarity is never achieved. Matters simply become muddier and muddier as we go down through time. Thus, far from providing the rungs of a ladder by which psychology gradually climbs to clarity, this form of conceptual structure leads rather to an ever increasing pile of issues, which we weary of or become diverted from, but never really settle. [27, pp. 2–7]

In the light of what Newell says, and as noted in Section 3.1.2, the reason that this kind of bottom-up strategy seems always to fail is that a theory that works in one local area rarely generalises to any other local area, or to any high-level view. Thus *a persistent focus on low-level observations and concepts, with little or no attention to high-level concepts, makes it difficult or impossible to achieve simplification and integration at high levels of abstraction.*

3.2.1 Hubert Dreyfus and climbing a tree to reach to moon

As early as the 1960s, Hubert Drefus was drawing attention to what he argued were serious weaknesses in much AI research at the time. A criticism that does seem to be as appropriate now as it was then, is the widespread assumption that research into some small aspect of human intelligence would eventually generalise into a general theory of human intelligence. In a striking analogy, Drefus suggested that the assumption is a bit like climbing a tree in the belief that this would somehow enable one to reach the moon [32, p. 119].

When many researchers adopt this assumption, and the assumption is still widespread, we get the fragmentation of AI into many subfields which is so prominent now.

3.2.2 Current AI is too narrow

In the same vein, Gary Marcus and Ernest Davis write:

> The central problem, in a word: current AI is *narrow*; it works for particular tasks that it is programmed for, provided that what it encounters isn't too different from what it has experienced before. That's fine for a board game like Go—the rules haven't changed in 2,500 years—but less promising in most real-world situations. Taking AI to the next level will require us to invent machines with substantially more flexibility. ... To be sure, ... narrow AI is certainly getting better by leaps and bounds, and undoubtedly there will be more breakthroughs in the years to come. But it's also telling: AI could and should be about so much more than getting your digital assistant to book a restaurant reservation. [33, pp. 12–14] (emphasis in the original)

3.3 The benefits of a top-down research strategy

The quotes in Sections 3.1 and 3.2 are, in effect, calls for a top-down, breadth-first strategy in AI research, developing a theory or theories that can be applied to a range of phenomena, not just one or two things in a narrow area.

Here are some key features of a top-down strategy in research, and their potential benefits:

1. *Broad scope.* Achieving generality requires that the data from which a theory is derived should have a broad scope, like the overarching goal of the SP programme of research, summarised at the beginning of Section 5.2.

2. *Ockham's razor, Simplicity and Power.* That broad scope is important for two reasons:

    - In accordance with Ockham's razor, a theory should be as *Simple* as possible but, at the same time, it should retain as much as possible of the descriptive and explanatory *Power* of the data from which the theory is derived (Appendix B).
    - But measures of Simplicity and Power are more important when they apply to a wide range of phenomena than when they apply only to a small piece of data—because the absolute gains in both Simplicity and Power are greater.

3. *If you can't solve a problem, enlarge it.* A broad scope, as above, can be challenging, but it can also make things easier. Thus Dwight D. Eisenhower is reputed to have said: 'If you can't solve a problem, enlarge it', meaning that putting a problem in a broader context may make it easier to solve. Good solutions to a problem may be hard to see when the problem is viewed through a keyhole, but become visible when the door is opened.

4. *Micro-theories rarely generalise well.* Apart from the potential value of 'enlarging' a problem (point 3 above), and broad scope (point 1), a danger of adopting a narrow scope is that, as noted in Section 3.2, any micro-theory or theories that is developed for one narrow area are

unlikely to generalise well to a wider context—with correspondingly poor results in terms of Simplicity and Power.

5. *Bottom-up strategies and the fragmentation of research.* The prevailing view about how to reach AGI seems to be '... that we'll get to general intelligence step by step by solving one problem at a time', expressed by Ray Kurzweil [34, p. 234]. But as noted in Section 3.2, much research in AI has been, and is to a large extent still working with this kind of bottom-up strategy: developing ideas in one area and hoping that they will generalise to other areas and eventually lead to AGI. But it seems that in practice the research rarely gets beyond two areas, and, as a consequence, there is much fragmentation of research.

3.4 The adoption of a top-down research strategy in the SP research

The overarching goal of the SP research is to simplify and integrate observations and concepts in AI, mainstream computing, mathematics, and human learning, perception, and cognition (Section 5.2).

In the quest for a general theory of those observations and concepts, the SPTI has been developed via a top-down, breadth-first research strategy with exceptionally wide scope. A clue was provided by the bioinformatics concept of 'multiple sequence alignment' which seemed to have the potential for the desired simplification and integration of concepts across a wide area (Section 5).

As described in Section 6.3, the concept of multiple sequence alignment led to the development of the concept of *SP-Multiple-Alignment*. Despite its similarity with the concept of multiple sequence alignment, a major programme of work was needed, as noted in Section 5.4, to develop the SP-Multiple-Alignment concept, including the creation and testing of *hundreds* of versions of the SPCM. This work included the exploration of a range of potential applications of the SP-Multiple-Alignment concept.

The SP strategy should help to meet the concerns of Gary Marcus and Ernest Davis: 'What's missing from AI today—and likely to stay missing, until and unless the field takes a fresh approach—is broad (or 'general') intelligence.' [33, p. 15].

4 The 'ICMUP' working hypothesis that information compression may always be achieved via the full or partial matching and unification of patterns

In view of the importance of IC in human learning, peception, and cognition (Section 2), it is clear that the SPTI, and any other theory of human-like intelligence with a central role for IC, must be broad enough to encompass several aspects of intelligence, and is consistent with the kinds of evidence described in Section 2.

With regard to the evidence outlined in Section 2 (the importance of IC in HLPC), the phenomena considered in Sections 2.3 (IC and probabilities), and 2.4 (IC and learning a first language), suggest the following working hypotheses:

- That 'redundancy' in any body of information, **I**, may be understood as the repetition of patterns in **I**, as described below.

- That it may be possible to understand all kinds of lossless IC of **I** in terms of reductions in redundancy in **I**.

- That reductions of redundancy in **I** may always be achieved via the merging or 'unification' of the repeating patterns of that redundancy. The expression 'IC via the matching and unification of patterns' may be abbreviated as 'ICMUP'.

There are four important qualifications of the idea that 'redundancy' in any body of information, **I**, may be understood as the repetition of patterns in **I**:

- Patterns that match each other need not be coherent groupings of symbols. For example, the SPCM can and often does work with partial matches between such SP-Patterns as '`t h r o w m e t h e b a l l`' and '`t h r o w d a d d y t h e b a l l`'.

- When compression of a body of information, **I**, is to be achieved via ICMUP, any repeating pattern that is to be unified should occur more often in **I** than one would expect by chance in a body of information of the same size as **I** that is entirely random. This may be referred to as the *Frequency Rule*.

  The point of the Frequency Rule is that, in ordinary English for example, there are normally many patterns that repeat but they are typically small patterns like '`ta`' or '`se`' that do not occur more frequently in **I** than one would expect by chance in a random text of the same size as **I**.

- Compression can be optimised by giving shorter codes to chunks that occur frequently and longer codes to chunks that are rare. This may be done using some such scheme as Shannon-Fano-Elias coding, described in, for example, [35].

- To be clear, the concept of 'pattern' in this context includes single symbols. Thus there may be redundancy in a body of information, **I**, because some symbols occur more frequently than one would expect by chance, even though **I** does not contain any redundancy in the form of patterns with two or more symbols that occur more frequently than one would expect by chance.

Although ICMUP is a 'working' hypothesis, there is much supporting evidence: the powerful SP-Multiple-Alignment concept (Section 6.3) may be understood as an example of ICMUP, and it is a generalisation of six other types of ICMUP (Section 4.2.7); the SP-Multiple-Alignment concept underpins the several intelligence-related strengths of the SPTI (Sections 8.1 and 8.2); and it appears that much of mathematics, perhaps all of it, may be understood in terms of ICMUP (Section 10).

4.1 Searching for repeating patterns

At first sight, the process of searching for repeating patterns is simply a matter of comparing one pattern with another to see whether they match each other or not. But in the SPTI, there may be matches between parts of two larger patterns, or there may be matches between patterns where the two patterns that match each other are each discontinuous within a larger pattern, such as a match between two instances of 'A B C' within 'p q A r B s t C u' and 'h A i j B k C l m'.

Thus 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, via unification, yield most compression—and a little reflection shows that this is not a trivial problem [1, Section 2.2.8.4].

To elaborate a little, maximising the amount of compression of **I** that may be achieved means maximising r where:

$$r = \sum_{i=1}^{i=n} (f_i - 1) \cdot s_i, \tag{2}$$

f_i is the frequency of the ith member of a set of n patterns within **I** and s is the size of that repeating pattern in bits. Patterns that are both big and frequent are best. This equation applies irrespective of whether the patterns are coherent substrings or, as noted above, 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 largest value. For an **I** containing m symbols, the number of possible subsequences (including single symbols and all composite patterns, both coherent and fragmented) is:

$$p = 2^m - 1. \tag{3}$$

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

$$c = p(p-1)/2. \tag{4}$$

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, we cannot normally guarantee to find the theoretically ideal answer, and normally we cannot know whether or not we have found the theoretically ideal answer.

In general, we need to use heuristic methods in searching (Appendix G)—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'.

There is more detail about finding good matches between two sequences of symbols in Appendix E.

4.2 Eight techniques for ICMUP

Once we have found 'good' matches between patterns, they may be encoded in terms of the following seven techniques for ICMUP, and possibly more that have not yet been identified. The seven techniques are fundamental in the SPTI and are central in the main thesis of [8], that, to a large extent, mathematics may be understood as ICMUP.

4.2.1 Basic ICMUP

The simplest of the techniques to be described is to find two or more patterns that match each other within a given body of information, **I**, and then merge or 'unify' them so that multiple instances are reduced to one. This is illustrated in the upper part of Figure 4 where two instances of the pattern

'INFORMATION' near the top of the figure has been reduced to one instance, shown just above the middle of the figure. Below it, there is the pattern 'INFORMATION', with 'w62' appended at the front, for reasons given in Appendix 4.2.2.

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

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

A detail that should not distract us from the main idea is that, in accordance with the Frequency Rule described in Section 4, when compression of a body of information, **I**, is to be achieved via Basic ICMUP, any repeating pattern that is to be unified should occur more often in **I** than one would expect by chance for a pattern of that size.

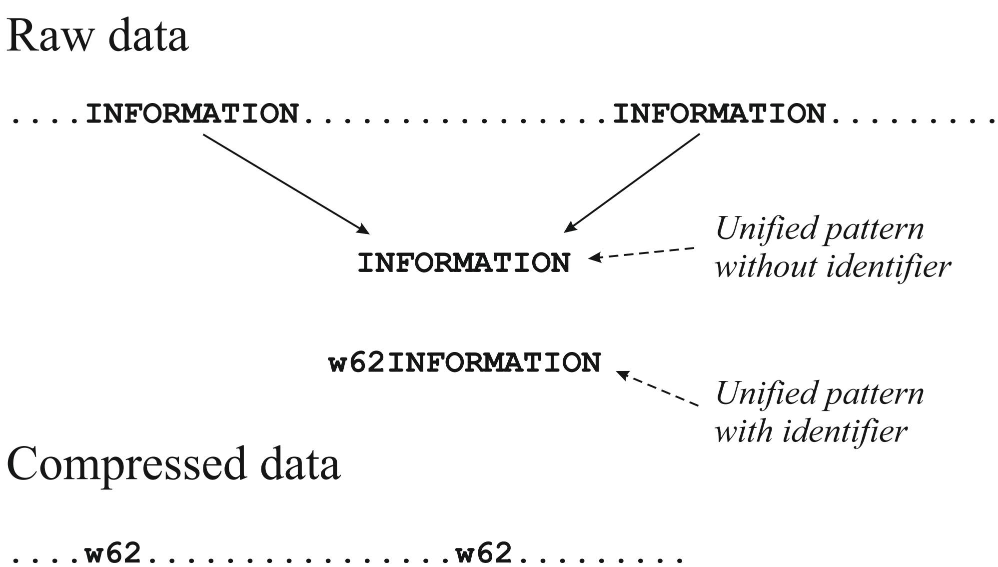

Figure 4: A schematic representation of the way two instances of the pattern 'INFORMATION' in a body of data may be unified to form a single 'unified pattern', shown just above the middle of the figure. To achieve lossless compression, the relatively short identifier 'w62' may be assigned to the unified 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 [1].

4.2.2 Chunking-With-Codes

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

This problem may be overcome with the chunking-with-codes variant of ICMUP, mentioned in Section 4.2.2 and described in more detail here:

- A unified pattern like '`INFORMATION`', which is often referred to as a 'chunk' of information,[5] is stored in a dictionary of patterns, as mentioned in Section 4.2.1.
- Now, the unified chunk is given a relatively short name, identifier, or 'code', like the '`w62`' pattern appended at the front of the '`INFORMATION`' pattern, shown below the middle of Figure 4.
- Then the '`w62`' code is used as a shorthand which replaces the '`INFORMATION`' chunk of information wherever it occurs within **I**. This is shown at the bottom of Figure 4.
- Since the code '`w62`' is shorter than each instance of the pattern '`INFORMATION`' which it replaces, the overall effect is to shorten **I**. But, unlike Basic ICMUP, chunking-with-codes may achieve 'lossless' compression of **I** because the original information may be retrieved perfectly at any time.
- Details here are:

    1. That compression can be optimised by giving shorter codes to chunks that occur frequently and longer codes to chunks that are rare.
    2. That, in accordance with the Frequency Rule, any chunk, **C**, to be given this treatment should be more frequent in **I** than the minimum needed (for a chunk of the size of **C**) to achieve compression (Section 4.2.1).

4.2.3 Schema-Plus-Correction

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

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

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

With this scheme, a particular meal may be represented economically as something like '`MN: ST(st2) MC(mc5) PG(pg3)`', where '`st2`' is the code or identifier for 'minestrone soup', '`mc5`' is the code for 'vegetable lassagne', and '`pg3`' is the code for 'ice cream'. Another meal may be represented economically as '`MN: ST(st6) MC(mc1) PG(pg4)`', where '`st6`' is the code or identifier for 'prawn cocktail', '`mc1`' is the code for 'lamb shank', and '`pg4`' is the code for 'apple crumble'; and so on. Here, the codes for different dishes serve as modifiers or 'corrections' to the categories '`ST`', '`MC`', and '`PG`' within the schema '`MN: ST MC PG`'.

4.2.4 Run-Length Coding

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

[5] There is a little more detail about the concept of 'chunk' in [11, Section 2.4.2].

‘INFORMATIONINFORMATIONINFORMATIONINFORMATIONINFORMATION’.

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

In a similar way, a sports coach might specify exercises as something like ‘touch toes (×15), push-ups (×10), skipping (×30), ...’ or ‘Start running on the spot when I say ‘start’ and keep going until I say ‘stop’ ’. With the ‘running’ example, ‘start’ marks the beginning of the sequence, ‘keep going’ in the context of ‘running’ means ‘keep repeating the process of putting one foot in front of the other, in the manner of running’, and ‘stop’ marks the end of the repeating process. It is clearly much more econonomical to say ‘keep going’ than to constantly repeat the instruction to put one foot in front of the other.

4.2.5 Class-Inclusion Hierarchies

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

This idea may achieve ICMUP because, at each level in the hierarchy, attributes may be recorded which apply to that level and all levels below it—so economies may be achieved because, for example, it is not necessary to record that cats have fur, dogs have fur, rabbits have fur, and so on. It is only necessary to record that mammals have fur and ensure that all lower-level classes and entities can ‘inherit’ that attribute. In effect, multiple instances of the attribute ‘fur’ have been merged or unified to create that attribute for mammals, thus achieving compression of information.[6]

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

4.2.6 Part-Whole Hierarchies

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

4.2.7 Generalisation of ICMUP via the sp-multiple-alignment concept

The seventh version of ICMUP, the *SP-Multiple-Alignment* concept described in Section 6.3, may be seen as a generalisation of all the preceding six versions of ICMUP [36].

[6]The Concept of Class-Inclusion Hierarchies with inheritance of attributes is quite fully developed in object-oriented programming, which originated with the Simula programming language [16] and is now widely adopted in modern programming languages.

The strengths of the SP-Multiple-Alignment concept in modelling aspects of human intelligence and the representation of intelligence-related knowledge is summarised in Sections 8.1 and 8.2, described fairly fully in [3], and much more fully in [1] with many examples of SP-Multiple-Alignments.

4.2.8 An eighth compression technique via 'objects'

The concept of an 'object' in computer programming, introduced in the Simula language [16] and now almost universal amongst programming languages, is not only useful as a means for programmers to model real-world objects, but may be understood as a mechanism for compressing information—a mechanism to be added to the SPTI and SPCM at some stage in the future.

Regarding the last point, the concept of an object may be seen as a version the concept of a 'chunk' of information (Appendix 4.2.2) but introduces a third dimension additional to the one dimension of SP-Patterns in the SPCM as it is now, or the two dimensions of SP-Patterns that may be incorporated in the SPCM in the future.

In future developments of the SPCM, objects may be abstracted from incoming data as described in [37, Sections 6.1 and 6.2].

4.3 Is ICMUP the same as 'symmetry'?

Superficially, ICMUP looks like the concept of 'symmetry' in physics, mathematics and other areas, because symmetry is at least partly about recognising similarities.

Noson Yanofsky and Mark Zelcer write [38, p. 1] that ''Symmetry' was initially employed in science as it is in everyday language.' and go on to describe how it has become relatively complex.

In general, ICMUP is distinctive in its great simplicity but more importantly in the way it provides the foundation for the concept of SP-Multiple-Alignment and its strengths in modelling aspects of intelligence and beyond (Section 8).

5 The origin and development of the sp theory of intelligence

Arriving at a good framework for the SPTI has not been straightforward. This section outlines how things progressed.

5.1 Early ideas

I first became interested in IC and its explanatory power from attending fascinating lectures about IC in the workings of brains and nervous systems, given by Dr Horace Barlow (later Professor Barlow FRS) at Cambridge University. Evidence for the importance of IC in human learning, peception, and cognition is described in [11].

The importance of IC in human cognition also became clear from my research,later, on the learning of a first language (Sections 1.1 and 2.4).

Another inspiration arose from browsing the book *Introduction to Theoretical Linguistics* by John Lyons [39]. Although this idea was not in the book, it occurred to me that the highly-structured nature of natural language, as described in the book, had something to say about how the human brain works. This idea has been a motivation and guide in my research.

Inspiration for the SPTI arose when I was employed in software development (Section 1.1) where there was often a the need for the integration of the diverse kinds of knowledge. This connected with the need for integration in the representation and processing of the syntax and semantics in the learning, interpretation, and production of natural language.

In that connection, there is a case for adopting a uniform system for the representation and processing of all kinds of knowledge. That principle has been adopted in the SPTI, as described in Sections 6.2 and 6.3.

5.2 Getting a grip on the problem

At some stage, it is not clear when, the project became one of developing a framework for the simplification and integration of observations and concepts across: AI; mainstream computing; mathematics; and human learning, peception, and cognition. This wide scope dictated a 'top down' approach to the research, seeking at the outset to identify a computational framework with wide scope (Section 3).

The first step was the creation of a program for finding good full and partial matches between two sequences of atomic symbols. This is described in [1, Appendix A]. The reasons for this choice were:

- *Information compression*:
  - Pioneering research by Fred Attneave [17, 18], Horace Barlow [**?**, 19], and others shows that much of the workings of brains and nervous systems may be understood as compression of information. That evidence, and much other evidence for the same conclusion, is described in [11].
  - It seemed likely that 'good' full and partial matches between sequences of symbols would also be ones that would allow compression of information via the merging or 'unification' of sections that match each other.
- Intuitively, it seemed likely that several aspects of intelligence might be understood as a search for good full and partial matches between patterns.
- *The potential for two or more good answers*:
  - Although, when this project began, many programs existed (and still exist) for finding good full and partial matches between sequences of symbols, it appeared that none of them could deliver more than a single best answer.
  - But intuition suggested that human intelligence normally entails the discovery of two or more answers of varying degree of success. Hence, the program that was developed in this programme of research was designed to find two or more reasonably good alternative solutions, or only one if alternative answers could not be found.

5.3 A key discovery: the potential of multiple sequence alignments or something like them

Having developed a program for finding good full and partial matches between two sequences of symbols, it seemed that there might be some value in looking at programs for finding good full and partial matches between two *or more* sequences of symbols. This kind of analysis, called 'multiple sequence alignment', is used by biochemists in the analysis of DNA sequences and in the analysis of sequences of amino-acid residues.

These programs are designed to analyse two, three, four, or more sequences simultaneously and to show each of the best results as an arrangement of the two or more sequences next to each other, with judicious 'stretching' of sequences in a computer to align symbols that match each other from one sequence to another.

An example of a 'good' multiple sequence alignment of five DNA sequences is shown in Figure 5.

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

Figure 5: A 'good' multiple sequence alignment amongst five DNA sequences. Reproduced from [1, Figure 3.1].

From studying this kind of multiple sequence alignment and creating different examples 'manually' with a word processor, I began to realise that multiple sequence alignments, or something like them, could provide a very versatile model for different aspects of AI.

5.4 Developing the SP-Multiple-Alignment concept

The concept of multiple sequence alignment was structured well enough to suggest how it might be useful in studies of human cognition or AI, but it was far from being usable in the embryonic SPTI.

To create a new SP-Multiple-Alignment concept for the understanding of intelligence, and for diverse applications of AI, required an extended period of development in which each idea about how the system might work was programmed into a version of the SPCM and then tested with appropriate data, across a range of potential areas of application.

For each idea, notes were taken, describing the idea and its successes or failures within a corresponding version of the SPCM, with ideas for how the original problems might be overcome, and the shortcoming in the model being addressed.

Although the SP-Multiple-Alignment concept as it is now (described in Section 6.3, below) may seem straightforward, its development took several years' work. The process of developing the SP-Multiple-Alignment concept required the creation and testing of *hundreds* of versions of the SPCM.

Apart from the SP-Multiple-Alignment concept itself, the process of developing the display of SP-Multiple-Alignments was a major challenge in which two-years' work was abandoned because an unworkable framework had been adopted.

6 The SP Theory of Intelligence and its realisation in the SP Computer Model

This section introduces the *SP Theory of Intelligence* and its realisation in the *SP Computer Model* with sufficient detail to ensure that the rest of the book is intelligible.[7] More detail may be found in the paper [3], and there is a much fuller account of the system in the book *Unifying Computing and Cognition* [1].

6.1 High level view of the SPTI

The SPTI is conceived as a brain-like system as shown in Figure 6, with *New* information (green) coming in via the senses (eyes and ears in the figure), and with some or all of that information compressed and stored as *Old* information (red), in the brain.

As described in more detail below, the processing of New information to create Old information is central in how the SPTI works, and it is central in all aspects of intelligence that are modelled by the SPTI, and apparently in all its potential applications.

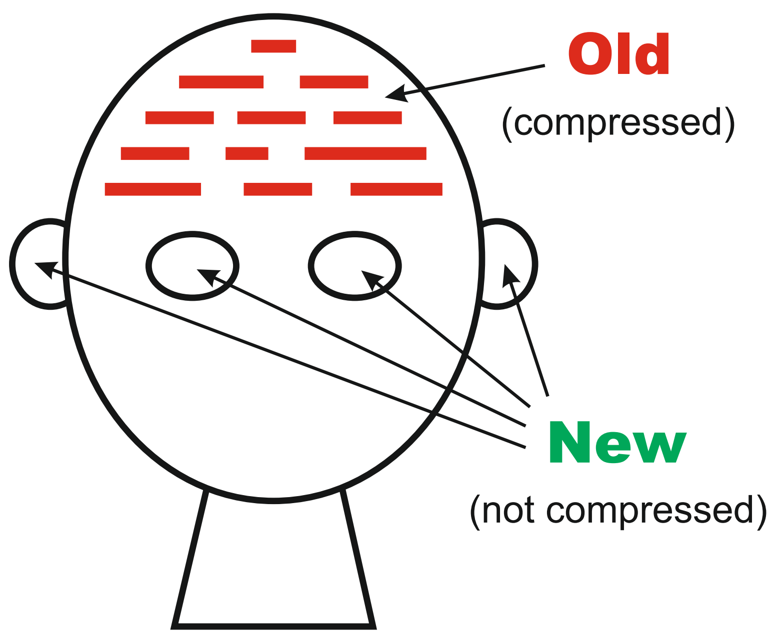

Figure 6: Schematic representation of the SPTI from an 'input' perspective. Reproduced from Figure 1 in [3].

[7]This section is based on [9, Appendix A].

6.2 *SP-Patterns* and *SP-Symbols*

In the SPTI, all kinds of knowledge are represented by *SP-Patterns*, where an SP-Pattern is an array of *SP-Symbols* in one or two dimensions.

An SP-Symbol is simply a mark from an alphabet of alternatives where each SP-Symbol can be matched in a yes/no manner with any other SP-Symbol.

An SP-Symbol does not have any hidden meaning, such as 'add' for the symbol '+' in arithmetic, or 'multiply' for the symbol '×', and so on. Any meaning attaching to an SP-Symbol is provided by one or more other SP-Symbols with which it is associated.

As noted in Appendix B, this kind of simplicity, combined with the relatively simple but powerful concept of SP-Multiple-Alignment (Section 6.3) is the key to the seamless integration of diverse aspects of AI and diverse kinds of intelligence-related knowledge, in any combination (Section 8.1.4).

Examples of SP-Patterns may be seen in Figure 7, as described in the caption to the figure.

6.2.1 Two-dimensional SP-Patterns

At present, the SPCM works only with one-dimensional SP-Patterns but it is envisaged that the SPCM will be generalised to work with two-dimensional SP-Patterns, as well as one-dimensional SP-Patterns. It is envisaged that 2D SP-Patterns may be of any shape, not necessarily simple rectangles

The addition of 2D SP-Patterns should open up the system for the representation and processing of diagrams and pictures. As described in [37, Sections 6.1 and 6.2)], 2D SP-Patterns may serve in the representation of structures in three dimensions. And 2D SP-Patterns may also have a role in the representation and processing of parallel streams of information, as described in [40, Sections V-G, V-H, and V-I, and Appendix C]. SP-Patterns in one or two dimensions may also serve in representing the time dimension in videos and the like.

There is more about the representation and processing of 2D patterns in Section 6.7.6.

6.2.2 ID SP-Symbols and C SP-Symbols

For some purposes, there is a distinction between 'ID' SP-Symbols and 'C' SP-Symbols. The former serve in the identification and classification of SP-Patterns, while the latter serve to represent the communicative contents of an SP-Pattern.

As an example, the SP-Pattern '`!N !Ns !4 w e s t !#N`' in the SP-grammar in Figure 8 (Section 6.3) contains the ID SP-Symbols '`N`', '`Ns`', '`4`', and '`#N`' at the ends of the SP-Pattern, and it contains the C SP-Symbols '`w`', '`e`', '`s`', and '`t`' in the middle of the SP-Pattern.

Notice that the character '`!`' at the beginning of each of the ID SP-Symbols '`N`', '`Ns`', '`4`', and '`#N`' serves to mark each SP-Symbol as an ID SP-Symbol in its current context but it is not part of the SP-Symbol. The same SP-Symbols may appear in other contexts without the preceding '`!`' character.

6.3 The *SP-Multiple-Alignment* concept

The SP-Multiple-Alignment concept is described in outline here. More detail may be found in [1, Section 3.4] and [3, Section 4].

6.3.1 The SP-Multiple-Alignment concept, Simplicity, and Power

The SP-Multiple-Alignment concept is largely responsible for the intelligence-related and other strengths of the SPTI, summarised in Section 8. Although it is far from being trivially simple, the SP-Multiple-Alignment concept is remarkably simple as the source of most of the strengths of the SPTI. In short, the relative Simplicity of the SPTI combined with its high descriptive and explanatory Power, is largely due to the SP-Multiple-Alignment concept.

In the light of the foregoing remarks, it is appropriate to say that *the SP-Multiple-Alignment concept is a major discovery with the potential to be as significant for an understanding of intelligence as is the concept of DNA for an understanding of biology. It may prove to be the 'double helix' of intelligence!*

6.3.2 Organisation of the SP-Multiple-Alignment concept

As we have seen (Section 5.3), the SP-Multiple-Alignment concept in the SPTI has been borrowed and adapted from the concept of 'multiple sequence alignment' in bioinformatics. An example of an SP-Multiple-Alignment is shown in Figure 7.

```
0                       t h e                p l u m      s                      a r e         r i p e           0
                        | | |                | | | |      |                      | | |         | | | |
1                       | | |          Nrt 6 p l u m #Nrt |                      | | |         | | | |           1
                        | | |          |             |    |                      | | |         | | | |
2                       | | |    N Npl Nrt           #Nrt s #N                   | | |         | | | |           2
                        | | |    | |                        |                    | | |         | | | |
3                  D 17 t h e #D | |                        |                    | | |         | | | |           3
                   |          |  | |                        |                    | | |         | | | |
4            NP 0a D          #D N |                        #N #NP               | | |         | | | |           4
             |                     |                            |                | | |         | | | |
5            |                     |                            |       V Vpl 11 a r e #V      | | | |           5
             |                     |                            |       |  |           |       | | | |
6            |                     |                            |  VP 1 V  |           #V A    | | | | #A #VP    6
             |                     |                            |  |       |              |    | | | | |   |
7 S Num    ; NP                    |                           #NP VP      |              |    | | | | |  #VP #S 7
     |     |                       |                                       |              |    | | | | |
8   Num PL ;                      Npl                                     Vpl             |    | | | | |         8
                                                                                          |    | | | | |
9                                                                                         A 21 r i p e #A        9
```

Figure 7: The best SP-Multiple-Alignment created by the SPCM that achieves the effect of parsing a sentence (‘t h e p l u m s a r e r i p e’) into its parts and sub-parts, as described in the text. The sentence in row 0 ia a New SP-Pattern, while each of the rows 1 to 9 contains a single Old SP-Pattern, drawn from a relatively-large repository of Old SP-Patterns, in the SP-grammar shown in Figure 8. Reproduced from Figure A2 in [9].

The main components of the SP-Multiple-Alignment concept, illustrated in the Figure 7, are these:

- Row 0 contains one New SP-Pattern representing information that has been received recently from the system's environment (Section 6.1). Sometimes row 1 contains more than one New SP-Pattern. In this case, the SP-Pattern in row 1 is a sentence but in other SP-Multiple-Alignments the New SP-Pattern may represent any other kind of information.

- Each of rows 1 to 9 contains a single Old SP-Pattern, drawn from a relatively large repository of Old SP-Patterns. In this case, that repository of Old SP-Patterns is the SP-grammar shown in Figure 8, and each Old SP-Pattern in the SP-Multiple-Alignment and in the SP-grammar represents a grammatical structure including words. More generally, Old SP-Patterns may represent any other kind of information.

In this example, the SP-Multiple-Alignment is shown with SP-Patterns in rows, but as we shall see, SP-Multiple-Alignments may also be shown with SP-Patterns in columns instead of rows (Figure 23, Section 7.4). The choice depends largely on what fits best on the page.

```
!S Num ; NP #NP VP !#VP #S
!NP !0 NP #NP Q #Q !#NP
!NP !0a D #D N #N !#NP
!VP !1 V #V A #A !#VP
!Q !1 P #P NP #NP !#Q
!N !Ns !2 i t !#N
!N !Ns !3 s h e !#N
!N !Ns !4 w e s t !#N
!N !Ns !5 o n e !#N
!Nrt !7 w i n d !#Nrt
!Nrt !6 p l u m !#Nrt
!N !Npl !8 s i x !#N
!N !Npl !9 s e v e n !#N
!N !Npl Nrt #Nrt s !#N
!V !Vs !10 d o e s !#V
!V !Vs !10a i s !#V
!V !Vpl !11 a r e !#V
!V !Vpl !12 p l a y !#V
!V !Vpl !13 d o !#V
!P !14 w i t h !#P
!P !15 f r o m !#P
!P !16 o f !#P
!D !17 t h e !#D
!D !18 a !#D
!A !19 s t r o n g !#A
!A !20 w e a k !#A
!A !21 r i p e !#A
!Num !SNG !; Ns Vs
!Num !PL !; Npl Vpl
```

Figure 8: In this SP-grammar, each SP-Pattern starts and ends with SP-Symbols representing a grammatical category which are called 'ID' SP-Symbols. The character '!' in the SP-grammar serves to mark a symbol as being an 'ID-symbol' (Section 6.2) and, in each SP-Pattern, the unmarked SP-Symbols are 'C' or 'contents' SP-Symbols.

6.3.3 How SP-Multiple-Alignments are built up

Here is a summary of how SP-Multiple-Alignments like the one shown in Figure 7 are built up:

1. At the beginning of processing, the SPCM has the afore-mentioned store of Old SP-Patterns (Figure 8).

   When the SPCM is more fully developed, those Old SP-Patterns would have been learned from raw data as outlined in Section 6.7, but for now they are supplied to the program by the user.

2. The next step is to read in the New SP-Pattern, '`t h e p l u m s a r e r i p e`', shown in row 0 of Figure 7.

3. Then the program searches for 'good' matches between the New SP-Pattern and Old SP-Patterns, where 'good' matches are ones that yield relatively high levels of compression of the New SP-Pattern in terms of Old SP-Pattern(s) with which it has been unified.

4. As can be seen in the figure, matches are identified at early stages between (parts of) the New SP-Pattern and the Old SP-Patterns '`D 17 t h e #D`', '`Nrt 6 p l u m #Nrt`', '`V Vpl 11 a r e #V`', and '`A 21 r i p e #A`'. Although this is not shown in this example, the SPCM also searches for matches *within* the New SP-Pattern.

5. In later stages, the program searches for matches between rows established in earlier stages. One of many examples in Figure 7 is the matches between rows 4 and 7.

6. In SP-Multiple-Alignments, IC may be achieved by collapsing the whole SP-Multiple-Alignment into a single sequence of symbols and thus unifying matching patterns within the SP-Multiple-Alignment, like the match between '`t h e`' in the New SP-Pattern and the same three letters in the Old SP-Pattern '`D 17 t h e #D`'. In practice, this is not done explicitly but only notionally to achieve a measure of IC for the whole SP-Multiple-Alignment and for each partial SP-Multiple-Alignment created in the course of building the whole SP-Multiple-Alignment.

7. The unification of '`t h e`' with '`D 17 t h e #D`' yields the unified SP-Pattern '`D 17 t h e #D`', with exactly the same sequence of SP-Symbols as the second of the two SP-Patterns from which it was derived.

8. Details of how IC for any one full or partial SP-Multiple-Alignment is calculated are given in Section 6.5.

9. As processing proceeds, similar pair-wise matches and unifications eventually lead to the creation of SP-Multiple-Alignments like that shown in Figure 7. At every stage, all the SP-Multiple-Alignments that have been created are evaluated in terms of IC, and then the best SP-Multiple-Alignments are retained and the remainder are discarded. In this case, the final 'winner' is the SP-Multiple-Alignment shown in Figure 7.

10. This process of searching for good SP-Multiple-Alignments in stages, with selection of good partial solutions at each stage, is an example of heuristic search, as described in Appendix G. As noted there, this kind of search is necessary because there are too many possibilities for

anything useful to be achieved by exhaustive search. By contrast, heuristic search can normally deliver results that are reasonably good within a reasonable computational complexity, but it cannot guarantee that the best possible solution has been found.

11. A simple but important detail is that any SP-Pattern in an SP-grammar, each one of which occurs only once in the SP-grammar, may appear two or more times in any SP-Multiple-Alignment, as may be seen in Figure 9, in which the SP=pattern '`NP D #D N #N #NP`' appears twice.

```
0          t h i s        b o y            l o v e s           t h a t        g i r l           0
           | | | |        | | |            | | | | |           | | | |        | | | |
1          | | | |        | | |            | | | | |           | | | |    N 0 g i r l #N        1
           | | | |        | | |            | | | | |           | | | |    |           |
2          | | | |        | | |            | | | | |    NP D   | | | | #D N           #N #NP    2
           | | | |        | | |            | | | | |    |  |   | | | | |                  |
3          | | | |        | | |            | | | | |    |  D 1 t h a t #D                 |     3
           | | | |        | | |            | | | | |    |                                 |
4          | | | |        | | |        V 0 l o v e s #V |                                 |     4
           | | | |        | | |        |             |  |                                 |
5 S NP     | | | |        | | |    #NP V             #V NP                               #NP #S 5
    |      | | | |        | | |     |
6   |  D 0 t h i s #D     | | |     |                                                           6
    |  |           |      | | |     |
7   NP D           #D N   | | | #N #NP                                                          7
                      |   | | | |
8                     N 1 b o y #N                                                              8
```

Figure 9: An SP-Multiple-Alignment showing how an SP-Pattern may appear more than once in the SP-Multiple-Alignment. In this case it is the SP-Pattern '`NP D #D N #N #NP`' which appears in row 7 and in row 2. Reproduced from [1, Figure 3.4 (a)].

6.3.4 Discontinuous constituents and their representation in an SP-Multiple-Alignment

Regarding the SP-Multiple-Alignment shown in Figure 7, an aspect not mentioned so far is the role of the SP-Pattern shown in row 8.

Clues to the role of that SP-Pattern lie in the SP-Symbols '`PL`', '`Npl`', and '`Vpl`' within the SP-Pattern in row 8. The first of these SP-Symbols, '`PL`', shows that the sentence has a 'plural' form. The second of those SP-Symbols, '`Npl`', marks the subject noun, '`p l u m s`' as having the plural form. And the third of those SP-Symbols, '`Vpl`', marks the main verb as having the plural form.

So in summary, the role of the SP-Pattern in row 8, which is the last SP-Pattern in the SP-grammar in Figure 8, is to encode the syntactic rule in English and many other natural languages that if the subject of the sentence has a plural form, then the main verb should also have a plural form.

As can be seen in the SP-grammar in Figure 8, the last but one SP-Pattern, '`!Num !SNG !; Ns Vs`', is the one that would be called into play to show the dependency between a singular subject noun and a singular main verb.

These dependencies are called 'discontinuous' because they can jump over any amount of intervening structure.

Arguably, this method for representing discontinuous dependencies in syntax is more elegant than the standard computer science method using variables, described in, for example, [41, Chapter 12]. The method illustrated in Figure 7 and described in [1, Section 5.4] has the merit of growing seamlessly out of the SP-Multiple-Alignment method for representing and processing linguistic information (amongst other kinds of information), without the need for any *ad hoc* addition to the method.

There is more detail about these kinds of discontinuous dependency in Sections 7.4 and 7.5.

6.4 Versatility of the SP-Multiple-Alignment concept

As noted in the caption to Figure 7, although the SP-Multiple-Alignment in the figure achieves the effect of parsing the sentence into its parts and sub-parts, *the beauty of the SP-Multiple-Alignment concept is that it is largely responsible for the versatility of the SPTI in intelligence-related areas* (summarised in Sections 8.1 and 8.2), *and in other areas less closely related to intelligence* (summarised in Section 8.3).

As noted in Section 4, the SP-Multiple-Alignment concept is the last of seven variants of ICMUP described there, and it has been shown to be a generalisation of the other six variants [36]. This generalisation is probably the main reason for the versatility of the SPTI described above.

6.5 Coding and the evaluation of an SP Multiple Alignment in terms of information compression

This section describes in outline how, in the SPTI, SP-Multiple-Alignments are evaluated in terms of IC. There is more detail in [3, Section 4.1] and [1, Section 3.5].

6.5.1 Preliminaries

Associated with each SP-Symbol type (meaning the name of all SP-Symbols that have the same name) is a notional *code* or bit pattern that serves to distinguish the given type from all the others. This is only notional because the bit patterns are not actually created. All that is needed for the purpose of evaluating SP-Multiple-Alignments is the size of the notional bit pattern associated with each SP-Symbol type.

The sizes of the codes are calculated (as described below) so that frequently-occurring SP-Symbols have shorter codes than SP-Symbols that occur more rarely. *Notice that these bit patterns and their sizes are totally independent of the sizes of the names for SP-Symbols used in written accounts like this one and chosen purely for their mnemonic value.*

Given an SP-Multiple-Alignment like the one shown in Figure 7, one can derive a *code SP-Pattern* from the SP-Multiple-Alignment in the following way:

1. Scan the SP-Multiple-Alignment from left to right looking for columns that contain an ID SP-Symbol (Section 6.2.2) that is not aligned with any other SP-Symbol.

2. Copy these unmatched ID SP-Symbols into a code SP-Pattern in the same order that they appear in the SP-Multiple-Alignment.

The code SP-Pattern derived in this way from the SP-Multiple-Alignment shown in Figure 7 is 'S PL 0a 17 6 1 11 21 #S'. This is, in effect, a compressed representation of those symbols in the New SP-Pattern that form hits with Old SP-Symbols in the SP-Multiple-Alignment.

In this case, the code SP-Pattern is a compressed representation of *all* the SP-Symbols in the New SP-Pattern. but it often happens that some of the SP-Symbols in the New SP-Pattern are *not* matched with any Old SP=symbols, and, in that case, the code SP-Pattern will represent only those New symbols that do form hits with Old symbols. The way in which the code SP-Pattern may be said to represent all or part of the New SP-Pattern is described in Section 6.5.4, below.

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

$$CD = b_n - b_e, \tag{5}$$

or a 'compression ratio,' as:

$$CR = b_n / b_e, \tag{6}$$

where b_n is the total number of bits in those SP-Symbols in the New SP-Pattern that form hits with Old SP-Symbols in the SP-Multiple-Alignment and b_e is the total number of bits in the code SP-Pattern ('encoding') that has been derived from the SP-Multiple-Alignment as described above.

CD and CR are each an indication of how effectively the New SP-Pattern (or those parts of the New SP-Pattern that form hits with Old SP-Patterns in the given SP-Multiple-Alignment) may be compressed in terms of the Old SP-Patterns that appear in the SP-Multiple-Alignment. The CD of an SP-Multiple-Alignment—which has been found to be more useful than CR—is often called the *compression score* of the SP-Multiple-Alignment.

In each of these equations, b_n is calculated as:

$$b_n = \sum_{i=1}^{h} c_i, \tag{7}$$

where c_i is the size of the code for ith symbol in a sequence, $h_1 ... h_j$, comprising those symbols within the New SP-Pattern that form hits with Old symbols within the SP-Multiple-Alignment.

b_e is calculated as:

$$b_e = \sum_{i=1}^{s} c_i, \tag{8}$$

where c_i is the size of the code for ith SP-Symbol in the sequence of s SP-Symbols in the code SP-Pattern derived from the SP-Multiple-Alignment.

6.5.2 Encoding individual SP-Symbols

The simplest way to encode individual SP-Symbols in New or Old SP-Patterns is with a 'block' code using a fixed number of bits for each SP-Symbol. But the SPCM uses variable-length codes for SP-Symbols, assigned in accordance with the Shannon-Fano-Elias coding scheme [35] so that the shortest codes represent the most frequent alphabetic SP-Symbol types and *vice versa*.

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

$$f_{st} = \sum_{i=1}^{p} (f_i \times o_i) \tag{9}$$

where f_i is the (notional) frequency of the ith SP-Pattern in the SP-grammar, o_i is the number of occurrences of the given SP-Symbol in the ith SP-Pattern and p is the number of SP-Patterns in the SP-grammar.

When the code sizes of each SP-Symbol type have been calculated, each SP-Symbol in the New SP-Pattern and each SP-Symbol in the set of Old SP-Patterns is assigned the code size corresponding to its type.

An important point to stress is that, when this process has been completed, the code size of each SP-Symbol in the New SP-Pattern is multiplied by a *cost factor*, normally about 2 or larger. This means that, in effect, each New SP-Symbol represents a relatively large chunk of information.

When a New SP-Symbol, with the size of its code multiplied by the cost factor, is aligned with a matching Old SP-Symbol, it may be encoded with the smaller code associated with the Old SP-Symbol, in much the same way that 'GMT' may be used as an abbreviation for 'Greenwich Mean Time' or 'WWW' is short for 'World Wide Web' in ordinary writing. This arrangement means that compression can occur at the level of individual SP-Symbols as well at the level of SP-Patterns. This has a bearing on the apparent paradox of 'decompression by compression', discussed in Section 8.3.3.

The reason for applying a cost factor to SP-Symbols in New SP-Patterns is to ensure that each New SP-Symbols is represented by a relatively large number of bits. This is to reflect the relatively large amounts of sensory information impinging on our eyes and ears, which, together with the relatively low channel capacity of the optic nerve, suggested to Barlow [**?**, p. 548] that IC is applied in the retina (Section 2.1.2), which is indeed the case.

There are many variations and refinements that may be made at the SP-Symbol level but, in general, the choice of coding system for individual SP-Symbols is not critical for the principles to be described below where the focus of interest is the exploitation of redundancy that may be attributed to sequences of two or more SP-Symbols rather than any redundancy attributed to individual SP-Symbols.

6.5.3 Analysis of a sentence

With the example shown in Figure 7, the New SP-Pattern is of course the sentence '`t h e p l u m s a r e r i p e`' shown in row 0.

If we follow the procedure for deriving an encoding from the SP-Multiple-Alignment described in Section 6.5.1, the result is the code SP-Pattern '`S PL 0a 17 6 1 11 21 #S`'. This means that the 15 SP-Symbols in the sentence may be encoded with the 9 SP-Symbols in the code SP-Pattern just shown.

This economy in terms of the SP-Symbols looks useful but not terribly dramatic. However, measuring savings in terms of the SP-Symbols used is really not appropriate. It makes much more sense to evaluate encodings in terms of the number bits of information that have been saved.

Here, we must take account of the cost factor which increases the number of bits associated with each New SP-Symbol, for the reason described in Section 6.5.2. To be more specific, the cost factor used in the creation of the SP-Multiple-Alignment in Figure 7 was 10. That meant that many more bits were needed to encode the letters in the sentence than were needed to encode the symbols in the code SP-Pattern for the sentence: an average of 779.50 bits for SP-Symbols in the sentence compared with an average of 60.82 bits for the Old SP-sybols that encode the sentence.[8]

[8]The fractional values in these two figures are simply the result of the methods of calculation described above.

6.5.4 Production of a sentence

As mentioned in Section 6.5.1, above, the way in which a code SP-Pattern may be said to represent all or part of a New SP-Pattern is described here. In brief, it means that the full or partial New SP-Pattern may be recreated by treating the code SP-Pattern as if it was a New SP-Pattern and processing it with the SPCM in exactly the same way as any New SP-Pattern.

This can be seen in Figure 10. All the words of the sentence '`t h e p l u m s a r e r i p e`', s' at the end of the word '`p l u m`' can be seen in the SP-Multiple-Alignment in the right order, thus recreating the whole sentence. Notice how, within the workings of the SPCM, individual code SP-Symbols serve to pick out the words or other structures with which they are associated: '`PL`' picks out the SP-Pattern for plural sentences (in row 8), '`0a`' picks out the SP-Pattern for noun phrases (in row 4), '`17`' picks out '`D 17 t h e #D`', and so on.

```
0 S     PL      0a   17                    6                          1       11            21                #S 0
  |     |       |    |                     |                          |       |             |                 |
1 S Num |  ; NP |    |                     |                   #NP VP |       |             |             #VP #S 1
     |  |  | |  |    |                     |                    |  |  |       |             |              |
2    |  |  | |  |    |                     |                    |  VP 1 V     |        #V A |          #A #VP    2
     |  |  | |  |    |                     |                    |       |     |        |  | |          |
3    |  |  | |  |    |                     |                    |       V Vpl 11 a r e #V | |          |         3
     |  |  | |  |    |                     |                    |          |              | |          |
4    |  |  | NP 0a D |        #D N         |                #N #NP         |              | |          |         4
     |  |  |       | |        |  |         |                |              |              | |          |
5    |  |  |       D 17 t h e #D |         |                |              |              | |          |         5
     |  |  |                     |         |                |              |              | |          |
6    |  |  |                     |     Nrt 6 p l u m #Nrt   |              |              | |          |         6
     |  |  |                     |      |             |     |              |              | |          |
7    |  |  |                     N Npl Nrt           #Nrt s #N             |              | |          |         7
     |  |  |                        |                                      |              | |          |
8   Num PL ;                       Npl                                    Vpl             | |          |         8
                                                                                          | |          |
9                                                                                         A 21 r i p e #A        9
```

Figure 10: The best SP-Multiple-Alignment created by the SPCM that achieves the effect of decoding the code SP-Pattern, '`S PL 0a 17 6 1 11 21 #S`', resulting in the recreation of the sentence '`t h e p l u m s a r e r i p e`', as described in the text. The SP-Pattern in row 0 ia a New SP-Pattern representing the encoding of the sentence, while each of the rows 1 to 9 contains a single Old SP-Pattern, drawn from the repository of Old SP-Patterns shown in Figure 8.

This process of reconstructing a sentence from its code may be seen as an example of what may appear paradoxically to be 'decompression by compression' described in Section 8.3.3, with a resolution of the paradox.

6.6 The calculation of absolute and relative probabilities for each SP Multiple Alignment

Regarding the calculation of absolute and relative probabilities of SP-Multiple-Alignments, this is done in the SPTI using information about the frequencies of occurrence of Old SP-Patterns, as outlined below. There is more detail in [1, Section 3.7] and [3, Section 4.4].

The formation of SP-Multiple-Alignments in the SP framework supports several kinds of probabilistic reasoning, as described in Section 8.1.2. The core idea is that any Old SP-Symbol, or group of Old SP-Symbols, in an SP-Multiple-Alignment that is *not* aligned with a New SP-Symbol or group of New SP-Symbols, represents an inference that may be drawn from the SP-Multiple-Alignment.

6.6.1 Absolute probabilities

Any sequence of t symbols, drawn from an alphabet of $|a|$types of SP-Symbol, represents one point in a set of n points where N is calculated as:

$$n = |a|^t. \tag{10}$$

if we assume that the sequence is random or nearly so, which means that the n points are equiprobable or nearly so, the probability of any one point (which represents a sequence of length t) is close to:

$$p_{ABS} = |a|^{-t}. \tag{11}$$

In the SPCM, the value of a is 2.

6.6.2 Is it reasonable to assume that New in encoded form is random or nearly so?

Why should we assume that the code for an SP-Multiple-Alignment is a random sequence or nearly so? In accordance with AIT (Section 1.3.1), a sequence is random if it is incompressible. If we have reason to believe that a sequence is incompressible or nearly so, then we may regard it as random or nearly so.

We cannot prove that no further compression of **I** is possible (unless **I** is very small). But we may say that, for a given set of methods and a given amount of computational resources that have been applied, no further compression can be achieved. In short, the assumption that the code for an SP-Multiple-Alignment is random or nearly so only applies to the best encodings found for a given body of information in New and must be qualified by the quality and thoroughness of the search methods which have been used to create the code.

6.6.3 Relative probabilities

The absolute probabilities of SP-Multiple-Alignments, 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.

A point we may note in passing is that the calculation of relative probabilities from p_{ABS} will tend to cancel out any general tendency for values of p_{ABS} to be too high or too low. Any systematic bias in values of p_{ABS} should not have much effect on the values which are of most interest to us.

If we are to compare one SP-Multiple-Alignment and its probability to another SP-Multiple-Alignment and its probability, *we need to compare like with like.* An SP-Multiple-Alignment can have a high value for p_{ABS} because it encodes only one or two symbols from New. It is not reasonable to compare an SP-Multiple-Alignment like that to another SP-Multiple-Alignment which has a lower value for p_{ABS} but which encodes more symbols from New. Consequently, the procedure for calculating relative values for probabilities (p_{REL}) is as follows:

1. For the SP-Multiple-Alignment which has the highest CD (which we shall call the *reference SP-Multiple-Alignment*), identify the symbols from New which are encoded by that SP-Multiple-Alignment. We will call these symbols the *reference set of symbols in New.*

2. Compile a *reference set of SP-Multiple-Alignments* which includes *the SP-Multiple-Alignment with the highest CD and all other SP-Multiple-Alignments (if any) which encode exactly the reference set of symbols from New, neither more nor less.*[9]

3. The SP-Multiple-Alignments 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.[10] Any SP-Multiple-Alignment which, after editing, matches another SP-Multiple-Alignment in the set is removed from the set.

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

$$p_{A\_SUM} = \sum_{i=1}^{i=R} p_{ABS_i} \tag{12}$$

where R is the size of the reference set of multiple alignments and p_{ABS_i} is the value of p_{ABS} for the ith multiple alignment in the reference set.

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

$$p_{REL_i} = p_{ABS_i} / p_{A\_SUM}. \tag{13}$$

6. Calculate the sum of the values for p_{ABS} in the reference set of SP-Multiple-Alignments:

$$p_{a\_SUM} = \sum_{i=1}^{i=r} p_{ABS_i} \tag{14}$$

where r is the size of the reference set of SP-Multiple-Alignments and p_{ABS_i} is the value of p_{ABS} for the ith SP-Multiple-Alignment in the reference set.

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

It is not necessary always to use the SP-Multiple-Alignment with the best CD as the basis of the reference set of symbols. It may happen that some other set of symbols from New is the focus of interest. In this case a different reference set of SP-Multiple-Alignments may be concepted and relative values for those SP-Multiple-Alignments may be calculated as described above.

[9] There may be a case for defining the reference set of SP-Multiple-Alignments as those SP-Multiple-Alignments which encode the reference set of symbols *or any super-set of that set*. It is not clear at present which of those two definitions is to be preferred.

[10] If Old is well compressed, this kind of redundancy amongst the rows of an SP-Multiple-Alignment should not appear very often.

6.6.4 Relative probabilities of patterns and symbols

It often happens that a given pattern from Old or a given alphabetic symbol type within patterns from Old appears in more than one of the SP-Multiple-Alignments in the reference set. In cases like these, one would expect the relative probability of the pattern or alphabetic symbol type to be higher than if it appeared in only one SP-Multiple-Alignment. To take account of this kind of situation, SPCM calculates relative probabilities for individual patterns and alphabetic symbol types in the following way:

1. Compile a set of patterns from Old, each of which appears at least once in the reference set of SP-Multiple-Alignments. No single pattern from Old should appear more than once in the set.

2. For each pattern, calculate a value for its relative probability as the sum of the p_{REL} values for the SP-Multiple-Alignments in which it appears. If a pattern appears more than once in an SP-Multiple-Alignment, it is only counted once for that SP-Multiple-Alignment.

3. Compile a set of alphabetic symbol types which appear anywhere in the patterns identified in step 2.

4. For each alphabetic symbol type identified in step 3, calculate its relative probability as the sum of the relative probabilities of the patterns in which it appears. If it appears more than once in a given pattern, it is only counted once.

With regard to alphabetic symbol types, the foregoing applies only to alphabetic types which do not appear in New. Any alphabetic type which appears in New necessarily has a probability of 1.0—because it has been observed, not inferred.

6.6.5 Comparison of SP-Multiple-Alignments that do not encode the same symbols from New

It is true that, when we compare SP-Multiple-Alignments, we should compare like with like in terms of the symbols from New which are encoded by the SP-Multiple-Alignment. But, nevertheless, there may be occasions when we wish to compare SP-Multiple-Alignments that do not encode the same symbols from New.

In cases like that, CD or CR can be used. It may be possible to develop a principled method for calculating probabilities of SP-Multiple-Alignments that occur in different subsets of the symbols in New but this has not, so far, been investigated.

6.7 Unsupervised Learning in the SP Theory of Intelligence

The main focus of learning in the SPTI is on 'unsupervised' learning, meaning learning without any kind of assistance from 'teachers', or the grading of data from simple to complex, or the like.

The main reasons for this focus on unsupervised learning are:

- The substantial body of evidence for the importance of IC in the workings of brains and nervous systems, including learning (Section 2), and the belief that the design of the SPTI should take advantage of what is known about human learning, peception, and cognition.

- Solomonoff's development of APT shows in outline how unsupervised learning may be achieved via IC (Appendix D).
- The weight of evidence that a child can learn his or her first language without the kinds of assistance mentioned above. For example, Christy Brown [42] learned English well enough for him to become the author of several books, despite the fact that his cerebral palsy meant that his speech was largely unintelligible, so there was little or no possibility for people to correct his language (see also [43]).
- Evidence from our own experience that, despite the existence of schools and colleges, most of human learning is unsupervised.
- The belief that unsupervised learning is the foundation for other kinds of learning such as learning by imitation, learning by being told, learning with rewards and punishments, and so on (Section 8.2.1).

In people, and in the SPTI, two different kinds of things can happen in learning, as described in the next two subsections.

As with the creation of SP-Multiple-Alignments, IC is central in unsupervised learning in the SPTI—and this is partly because SP-Multiple-Alignments play an important part within unsupervised learning in the SPTI.

6.7.1 Learning wHen New information matches something in the store of old knowledge

As an example of learning when there is already some knowledge in the store of Old knowledge, Figure 11 (a) shows partial matching in an SP-Multiple-Alignment between one New SP-Pattern and one Old SP-Pattern.

```
0     t h a t g i r l r u n s    0
      | | | |         | | | |
1 A 1 t h a t b o y   r u n s #A 1

(a)

B 2 t h a t #B
C 3 b o y #C
C 4 g i r l #C
D 5 r u n s #D
E 6 B #B C #C D #D #E

(b)
```

Figure 11: (a) A simple SP-Multiple-Alignment from which, via the SPTI, Old patterns may be derived. (b) Old SP-Patterns derived from the SP-Multiple-Alignment shown in (a).

From a partial matching like this, the SPCM derives SP-Patterns that reflect coherent sequences of matched and unmatched SP-Symbols, and it stores the newly-created SP-Patterns in its repository of Old SP-Patterns. The results in this case are shown in Figure 11 (b).

There are two important features of this learning:

- Each of the words shown in Figure 11 (b) has SP-Symbols added at the beginning and end like '`B`', '`2`', '`#B`' in the SP-Pattern '`B 2 t h a t #B`'. These SP-Symbols are the 'identification' SP-Symbols, or '`ID`' SP-Symbols introduced in Section 6.2.2.
- The SP-Pattern '`E 6 B #B C #C D #D #E`' records the order of the words in Figure 11 (a). Notice in particular that the words '`C 3 b o y #C`' and '`C 4 g i r l #C`' are shown as alternatives in the middle of the structure because they both begin with '`C`' and end with '`#C`'.

As the learning process proceeds, it builds up alternative structures, each one comprising an *SP-grammar*, **G**, together with an SP-encoding, **E**, of the New information in terms of **G**. Here, an SP-grammar is simply a set of Old SP-Patterns that is relatively good at compressing a given set of New SP-Patterns.

In terms of that compression, some SP-grammars are better than others. In the SPTI, there is a process of heuristic search (Appendix G) which retains the relatively good SP-grammars, and discards the rest.

In accordance with the principles of APT (Appendix D), the aim of these processes of heuristic search is to minimise $(g + e)$, where g is the size (in bits) of each full or partial SP-grammar **G** that has been created and e is the size (in bits) of the encoding **E** of the New pattern in terms of **G**. Here, **E** is constructed as described in Section 6.5.

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

$$g = \sum_{i=1}^{i=g} (\sum_{j=1}^{j=k_i} s_j) \tag{15}$$

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

In the SPTI, the processes just described are the essentials of unsupervised learning of SP-grammars.

6.7.2 Learning with a *tabula rasa* or when New information does not match anything in the repository of Old information

With people, the closest we come to learning as a *tabular rasa*—learning with nothing in our memories—is when we are babies, and even then we have some inborn knowledge. But much the same happens in the SPTI when New information does not match anything in the repository of Old information.

In either case—learning when the SPTI is a *tabula rasa* or learning when New information matches nothing in the store of Old information—the SPCM learns by taking in New SP-Patterns via its 'senses' and storing them directly as received, except that ID SP-Symbols are added at the beginning and end of each SP-Pattern, like the SP-Symbols '`A`', '`21`', and '`#A`', in the SP-Pattern '`A 21 r i p e #A`' in row 9 of Figure 7. As mentioned earlier, those added SP-Symbols provide the means of identifying and classifying SP-Patterns.

A qualification to the paragraph immediately above is that, as noted in Section 6.3, in addition to comparing New information with Old information, the SPCM searches for redundancy *within* each New SP-Pattern and reduces or eliminates any such redundancy wherever it is found.

The kind of direct learning of New information described above reflects the way that people may learn from a single event or experience [6, Section 7]. One experience of getting burned may teach a child to take care with hot things, and the lesson may stay with him or her for life. Also, we often remember quite incidental things from one experience that have no great significance in terms of pain or pleasure—such as a glimpse we may have had of a red squirrel climbing a tree.

Any or all of this one-shot learning may go into service immediately without the need for repetition, as for example: when we ask for directions in a place that we have not been to before; or how, in a discussion, we normally respond to what other people are saying.

These kinds of one-shot learning contrast sharply with:

- Learning in DNNs which requires large volumes of data and many repetitions before anything useful is learned. In that connection, Yann LeCun writes [34, p. 124]):

  > ... the first time you train a convolutional network you train it with thousands, possibly even millions of images of various categories.

- The model of language learning proposed by Gold [25], where learning is not possible without negative examples (examples that are marked as 'wrong') or correction of errors by a 'teacher', or the grading of language samples from simple to complex.

6.7.3 An example

When New contains the eight sentences shown in Figure 12, the best grammar found by thr SPCM is the one shown in Figure 13.

```
t h a t b o y r u n s
t h a t g i r l r u n s
t h a t b o y w a l k s
t h a t g i r l w a l k s
s o m e b o y r u n s
s o m e g i r l r u n s
s o m e b o y w a l k s
s o m e g i r l w a l k s
```

Figure 12: Eight sentences supplied to the SPCM as New SP-Patterns.

```
< %2 2 s o m e >
< %2 3 t h a t >
< %1 5 b o y >
< %1 6 g i r l >
< %3 4 r u n s >
< %3 7 w a l k s >
< 1 < %2 > < %1 > < %3 > >
```

Figure 13: The best SP-grammar found by the SPCM when New contains the eight sentences shown in Figure 12.

This result looks reasonable but one may wonder why the terminal 's' of 'r u n s' and 'w a l k s' has not been identified as a discrete entity, separate from the verb stems 'r u n' and 'w a l k'.

In the pattern-generation phase of processing, the SPCM does form multiple alignments like this

```
0         t h a t b o y w a l k s   0
          | | | | | | |         |
1 < %1 9 t h a t b o y r u n    s > 1
```

which clearly recognises the verb stems and 's' as distinct entities. But for reasons that are still not clear, the program does not build these entities into plausible versions of the full sentence structure.

6.7.4 How, in unsupervised learning, to make generalisations without over- or under-generalisation

As Chomsky [44] and others have argued cogently, an adult's knowledge of his or her native language is much more general than the large but finite sample that he or she has heard since birth. From an early age children show signs of creating general rules such 'Add *ed* to a verb to give it a past tense' or 'Add *s* to a noun to make it plural' and, in the early stages, they often overgeneralise such rules and say such things as 'Mummy buyed it', 'There are some sheeps' and so on. Of course they learn to correct such overgeneralisations, apparently without the need for explicit error correction by adults or older children.

In brief, the problem to be solved with unsupervised learning is how, without any kind of 'teacher' or correction of errors by anyone else, a child in learning his or her first language, can generalise beyond the language that they hear and can correct over-generalisations (under-fitting) and under-generalisations (over-fitting).

The learning problem may be represented schematically as shown in Figure 14. The smallest envelope shows the set of 'utterances' that constitute the finite sample of utterances from which a grammar is to be inferred. The middle-sized envelope represents the (infinite) set of utterances in the language being learned. And the largest envelope represents the (infinite) set of all possible utterances. As discussed in Section 6.7.5, children also have to cope with dirty data meaning 'wrong' utterances that are part of the sample from which they learn.

The solution proposed by Solomonoff and described in Appendix D is IC applied to the body of data that is the basis for learning, which for people means all the language which the learner has heard since birth.

In terms of the SPTI concepts, Solomonoff's solution may be expressed like this:

- The data to be compressed is New information received via the system's 'senses' and the result of compression is the Old information stored in the system's 'brain' (Section 6.1).
- For each sentence or other item of information in New, compression is achieved by creating a code for the sentence, as described in Section 6.5.
- IC is achieved by creating an SP-grammar **G** for the New information, meaning a set of SP-Patterns where each SP-Pattern occurs two or more times in the New information, together with an encoding **E** of the New information in terms of **G**.
- In general, the amount of compression that is achieved is not any kind of theoretical ideal but is the amount of compression that can be achieved with a 'reasonable' amount of processing, meaning whatever is available when other demands have been taken into account.

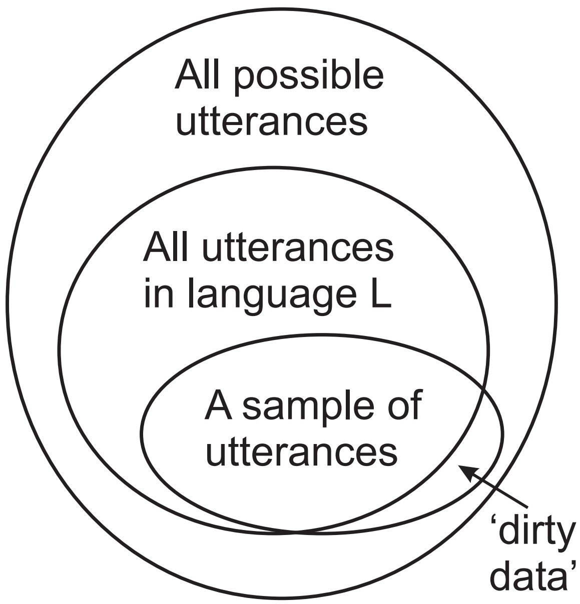

Figure 14: Categories of utterances involved in the learning of a first language, **L**. In ascending order size, they are: the finite sample of utterances from which a child learns; the (infinite) set of utterances in **L**; and the (infinite) set of all possible utterances. Adapted from Figure 7.1 in [24], with permission.

- In those terms, the general aim is to minimise $(g + e)$, where g is the size **G** and e is the size of **E**.

6.7.5 How, in unsupervised learning, to minimise the corrupting effect of dirty data

The question addressed in this section is how to learn correct forms despite the fact that **I** normally contains errors of various kinds, otherwise called dirty data (shown in Figure 14).

In previous writings about the SPTI, the proposal for minimising dirty data was wrong. As previously proposed, the solution was to minimise $(g + e)$ and then discard **E** which may of course discard useful information.[11]

A better answer now appears to be this: minimise $(g + e)$ and then discard those parts of New that have not been incorporated in either **G** or **E**. In effect this means discarding anything that appears only once in New. This is likely to include slips of the tongue, half-completed sentences, and the like. But it may include valid information that just happens to be rare.

The justification for this method of dealing with dirty data is roughly the same as the rule adopted by the more professional news media: that, in general, they should only report events that have been confirmed, meaning evidence from at least two sources.

With 'dirty data' that has been discarded from one body of New information, it is possible of course to add it to a later body of New information, in which case it may gain confirmation later.

6.7.6 Learning structures in two, three and four dimensions

At present, the SPCM is limited to the representation and processing of structures in one dimension but it may be developed for the representation and processing of structures in 2, 3, and 4 dimensions [2, Section 9], where the fourth dimension is the time dimension in videos and the like.

The Unsupervised Learning of 2D patterns Although the SPCM as it is now (with 1D SP-Patterns) may in principle be developed for the representation of 2D patterns, that development will be greatly facilitated when the SPCM has been generalised to work with 2D SP-Patterns (Section 6.2).

Although two-dimensional SP-Patterns can provide a basis for the representation of 2D structures, it is likely that all such structures in their 'raw' state will contain redundancies in areas that are uniform, surrounded by borders where information is concentrate (Section 2.1.1), and that, information 'is further concentrated at those points on a contour at which its direction changes most rapidly' [17, p. 184].

The Unsupervised Learning of 3D patterns The learning of structures in two dimensions would open the door for the learning of 3D structures as described in [37, Sections 6.1 and 6.2] and outlined here.

In brief, if an object is viewed from several different angles, with overlap between one view and the next (as illustrated in Figure 15), the several views may be stitched together to create what is at least a partial and approximate 3D model of the object, in much the same way that a panoramic photo may be created from a sequence of overlapping shots. This kind of processing

[11] Also, in previous publications, such as [6, p. 1073] and [1, Section 2.2.12], I have wrongly credited Solomonoff with this (wrong) solution to the dirty data problem. He did not address the problem at all.

may be achieved via an SP-Multiple-Alignment that accommodates 2D SP-Patterns and has been generalised to find redundancies within and between SP-Patterns.

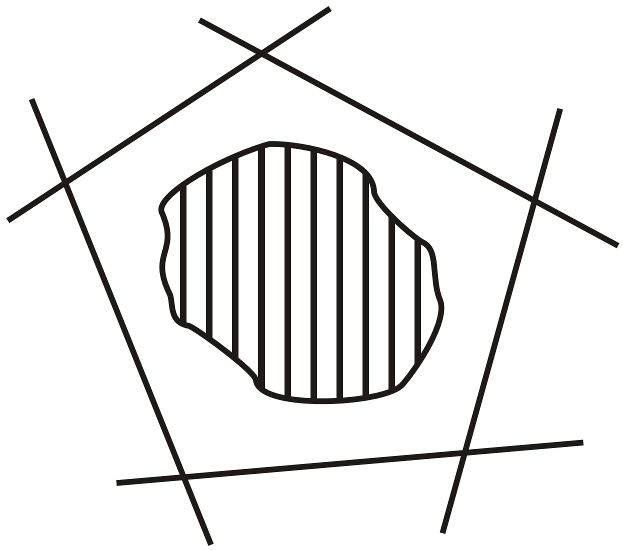

Figure 15: Plan view of a 3D object, with each of the five lines around it representing a view of the object, as seen from the side. Reproduced from [37, Figure 11].

The model will be partial if, for example, it excludes views from above or below. And it is likely to be approximate because a given set of views may not be sufficient for an unambiguous definition of the object's geometry: there may be variations in the shape that would be compatible with the given set of views.

Do these deficiencies matter? For many practical purposes, the answer is likely to be 'no'. If we want a rock to put in a rockery, or a stick to throw for a dog, the exact shape is not important. And if we want more accurate information, we can inspect the object more closely, or supplement vision with touch.

The Unsupervised Learning of 4D patterns Last but not least, the fourth or time dimension is of course a major feature of videos and films and must be accommodated in future versions of the SPTI [2, Section 9.3]. And most videos and films will give great scope for ICMUP because a typical frame is very similar to the one before it, and to the one that follows it.

6.7.7 Plotting values for o, g, e and t

To provide a more rounded picture of unsupervised learning in the SPTI, Figure 16 shows cumulative values for critical variables as the 8 sentences shown in Figure 12 (Section 6.7.3) are processed, one at a time. The variables are, at each of the 8 stages: o meaning the size of **O** which is the sentences being processed; g which is the size of **G**, the best SP-grammar derived from the sentences; e which is the size of **E**, the encodings of the sentences in terms of the best SP-grammar; t which is the sum of g and e.

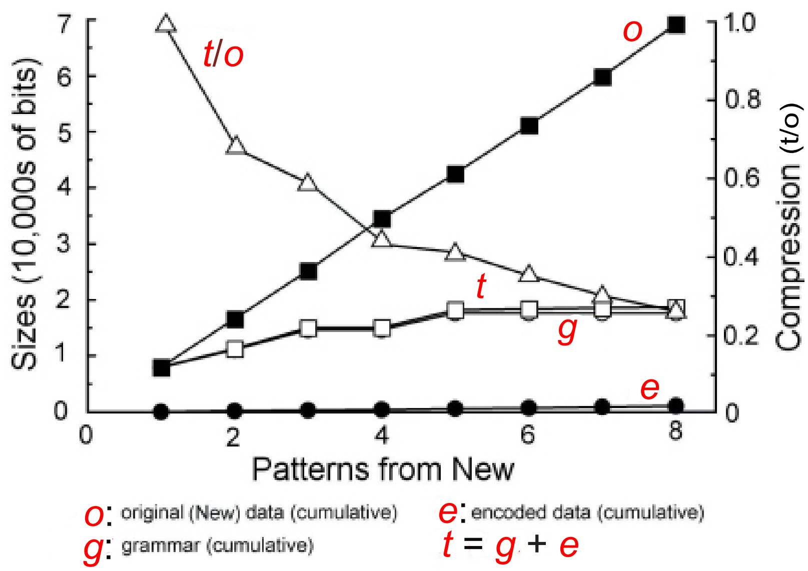

Figure 16: Changing values for the sizes of o, g, e and t and related variables as SPCM learning proceeds, with New SP-Patterns shown in Figure 12. Adapted from Figure 9.12 in [1].

As one might expect, the graph for the cumulative value of o rises steadily as each sentence is added to the pool of sentences. The graphs for g and for t rise much more slowly showing that the compression is effective. Correspondingly, the graph for the cumulative values for t divided by the cumulative values of o falls steadily from left to right.

6.7.8 Computational complexity

In common with other programs for unsupervised learning (and, indeed, other programs for finding good multiple alignments), the SPCM does not attempt to find theoretically ideal solutions.[12] This is because the abstract space of possible grammars (and the abstract space of possible multiple alignments) is, normally, too large to be searched exhaustively. In general, heuristic techniques must be used (Appendix G). By using these techniques, one can normally convert an intractable computation into one with a computational complexity that is within acceptable limits.

In the SPCM, the critical operation is the formation of SP-Multiple-Alignments. Other operations are, in comparison, quite trivial in their computational demands.

In a serial processing environment, the time complexity of the *create-sp-multiple-alignments()* function is $O(\log_2 n \times nm)$ [45], where n is the size of the pattern from New (in bits) and m is the sum of the lengths of the patterns in Old (in bits). In a parallel processing environment, the time complexity may approach $O(\log_2 n \times n)$, depending on how well the parallel processing is applied. In serial and parallel environments, the space complexity is $O(m)$.

[12] This section is based on [1, Section 9.3.1]

In the SPCM, the function is applied (twice) to the set of patterns in New so we need to take account of how many patterns there are in New. It seems reasonable to assume that the sizes of patterns in New are approximately constant.[13]

Old is initially empty and grows as learning proceeds. The size of Old (before purging) is, approximately, a linear function of the size of New. Given this growth in the size of Old, the time required to create multiple alignments for any given pattern from New will grow as learning proceeds. Again, the relationship is approximately linear.

So if we ignore operations other than the *create-sp-multiple-alignments()* function, the time complexity of the program (in a serial environment) is $O(N^2)$ where N is the number of patterns in New. In a parallel processing environment, the time complexity may approach $O(N)$, depending on how well the parallel processing is applied. In serial or parallel environments, the space complexity is $O(N)$.

6.8 The SP Computer Model

The SPTI is realised most fully in the SP Computer Model, with capabilities in the building of SP-Multiple-Alignments and in unsupervised learning. The source code for the SPCM (the current version of which is 'SP71'), with a Windows executable file and some other files, may be obtained via sources detailed in 'Software Availability', after the Conclusion (Section 14).

For reasons outlined in Section 1.3.4, the SPCM and its precursors have played a key part in the development of the SPTI:

- As an antidote to vagueness. As with all computer programs, processes must be defined with sufficient detail to ensure that the program actually works.
- By providing a convenient means of encoding the simple but important mathematics that underpins the SPTI, and performing relevant calculations, including calculations of probability.
- By providing a means of seeing quickly the strengths and weaknesses of proposed mechanisms or processes. Many ideas that looked promising have been dropped as a result of this kind of testing.
- By providing a means of demonstrating what can be achieved with the theory.

The workings of the SPCM is described in some detail in [1, Sections 3.9, 3.10, and 9.2] and more briefly in Sections 6.3 and 6.7, above.

6.9 Things to be done

This section describes things that need doing to bring things forward with the SPTI, firstly filling gaps in what has been done to date (Section 6.9.1), and secondly looking forward to how the system may be developed in the future (Section 6.9.2).

[13] There is no requirement in the model that patterns in New should, for example, be complete sentences. They may equally well be arbitrary portions of incoming data, perhaps measured off by some kind of input buffer.

6.9.1 Unfinished business

Like most theories, the SPTI has shortcomings, but it appears that they may be overcome. At present, the most immediate problems are:

- *Processing of information in two or more dimensions.* No attempt has yet been made to generalise the SP model to work with patterns in two dimensions, although that appears to be feasible to do, as outlined in [1, Section 13.2.1]. As noted in [1, Section 13.2.2], it is possible that information with dimensions higher than two may be encoded in terms of patterns in one or two dimensions, somewhat in the manner of architects' drawings. A 3D structure may be stitched together from several partially overlapping 2D views, in much the same way that, in digital photography, a panoramic view may be created from partially overlapping pictures Section 6.7.6.

- *Recognition of Perceptual Features in Speech and Visual Images.* For the SPTI to be effective in the processing of speech or visual images, it seems likely that some kind of preliminary processing will be required to identify low level perceptual features such as, in the case of speech, phonemes, formant ratios, or formant transitions, or, in the case of visual images, edges, angles, colours, luminances, or textures. In vision, at least, it seems likely that the SP framework itself will prove relevant since edges may be seen as zones of non-redundant information between uniform areas containing more redundancy and, likewise, angles may be seen to provide significant information where straight edges, with more redundancy, come together ([17, pp. 184–185], [37, Section 3]). As a stop-gap solution, the preliminary processing may be done using existing techniques for the identification of low-level perceptual features [46, Chapter 13].

- *Unsupervised learning.* A limitation of the SP computer model as it is now is that it cannot learn intermediate levels of abstraction in grammars (e.g., phrases and clauses), and it cannot learn the kinds of discontinuous dependencies in natural language syntax that are described in [3, Sections 8.1 and 8.2].

- *Processing of numbers.* The SP model works with atomic symbols such as ASCII characters or strings of characters with no intrinsic meaning. In itself, the SPTI does not recognise the arithmetic meaning of numbers such as '37' or '652' and will not process them correctly. However, the system has the potential to handle mathematical concepts if it is supplied with patterns representing Peano's axioms or similar information [1, Chapter 10]. As a stop-gap solution, existing technologies may provide whatever arithmetic processing may be required.

I believe these problems are soluble and that solving them will greatly enhance the capabilities of the system for the unsupervised learning of structure in data [3, Section 5.1].

In the process of solving these and other problems in the development of the SPTI, it seems likely that the proposed *SP Machine* (Section 6.9.2, next) will be a useful vehicle for the representation and testing of ideas.

6.9.2 Future developments and the SP Machine

In view of the potential of the SPTI in diverse areas (Sections 8.1 and 8.2), the SPCM appears to hold promise as the foundation for the development of an industrial-strength *SP Machine*, described in [2], and illustrated schematically in Figure 17.

It is envisaged that the SP Machine will feature high levels of parallel processing and a good user interface. It may serve as a vehicle for further development of the SPTI by researchers anywhere. Eventually, it should become a system with industrial strength that may be applied to the solution of many problems in science, government, commerce, industry, and in non-profit endeavours.

It is envisaged that the best way forward is to develop the *SP Machine* by porting the SPCM on to a platform which will provide for the application of high levels of parallel processing, and to adapt the SPCM to exploit those high levels of parallel processing. Additionally, there is a need to give the system a good 'friendly' user interface.

Although it is likely that a mature version of the SP Machine will be very much more efficient than the extraordinarily power-hungry and data-hungry DNNs [6, Section 9], high levels of parallel processing are likely to be needed for relatively demanding operations in the SPTI such as unsupervised learning, especially with 'big data' and the like.

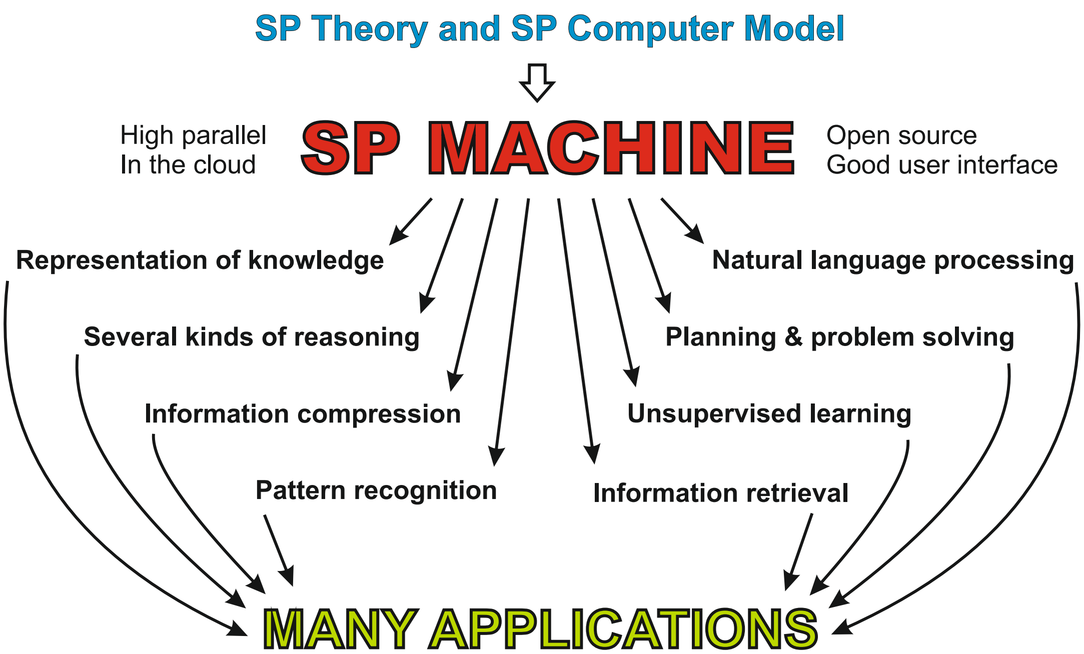

Figure 17: Schematic representation of the development and application of the SP Machine. Reproduced from Figure 2 in [3].

It is envisaged that the design and development of the SP Machine will be entirely open so that researchers anywhere may test the system and help to develop it, perhaps following the suggestions in [2]. To make things easy for other researchers, the SP Machine may be hosted on one or more of the following platforms:

- A workstation with GPUs providing high levels of parallel processing. Other researchers would need to buy one or more such workstations, and then, on each machine, they may install the open-source software of the SPCM, ready for further development.

- Facilities in the cloud that provide for high levels of parallel processing.
- Since pattern-matching processes in the foundations of the SPCM are similar to the kinds of pattern matching that are fundamental in any good search engine, an interesting possibility is to create the SP Machine as an adjunct to one or more search engines. This would mean that, with search engines that are not open access, permission would be needed to access functions in relevant parts of the search engine, so that those functions may be used within the SP Machine.

7 Examples of the versatility of the SP-Multiple-Alignment concept within the SPCM

As the title of this section suggests, it contains examples of SP-Multiple-Alignments showing the kinds of things that may be done with the SPCM. These examples show some of the versatility of the SP-Multiple-Alignment concept, but SP-Multiple-Alignments are very much more versatile than these few examples may suggest.

7.1 Recursive processing and the SP Theory of Intelligence

This subsection shows, with the recognition of a palindrome as an example, how the SPCM may accommodate recursive processing. There are two other examples of recursive processing in Figure 53 in Section 13.1.4.

Regarding the recognition of a palindrome, Figure 19 shows the best SP-Multiple-Alignment produced by the SPCM with 'a c b a b a b c a' in New and, in Old, the SP-Patterns under the heading 'Old' in Figure 18. The SP-Multiple-Alignment may be seen as a recognition that the pattern in New is indeed a palindrome.

New

a c b a b a b c a

Old

L a #L
L b #L
L c #L
L1 a #L1 L2 a #L2
L1 b #L1 L2 b #L2
L1 c #L1 L2 c #L2
X L #L #X
X L1 #L1 X #X L2 #L2 #X

Figure 18: SP-atterns for processing by the SPCM to model the recognition of a palindrome.

```
0       a          c          b          a         b          a            b           c           a         0
        |          |          |          |         |          |            |           |           |
1       |          |          |       L1 a #L1     |       L2 a  #L2       |           |           |         1
        |          |          |       |     |      |       |      |        |           |           |
2       |          |          |     X L1   #L1 X   |    #X L2    #L2 #X    |           |           |         2
        |          |          |     |          |   |    |            |     |           |           |
3       |          |       L1 b #L1 |          |   |    |            |  L2 b #L2       |           |         3
        |          |       |     |  |          |   |    |            |  |     |        |           |
4       |          |     X L1   #L1 X          |   |    |            #X L2   #L2 #X    |           |         4
        |          |     |                     |   |    |                        |     |           |
5       |       L1 c #L1 |                     |   |    |                        |  L2 c #L2       |         5
        |       |     |  |                     |   |    |                        |  |     |        |
6       |     X L1   #L1 X                     |   |    |                        #X L2   #L2 #X    |         6
        |     |                                |   |    |                                    |     |
7   L1 a #L1 |                                |   |    |                                    |  L2 a #L2     7
     |     |  |                                |   |    |                                    |  |     |
8 X L1   #L1 X                                |   |    |                                    #X L2   #L2 #X  8
                                               |   |    |
9                                              | L b #L |                                                    9
                                               | |   |  |
10                                             X L   #L #X                                                  10
```

Figure 19: The best SP-Multiple-Alignment (in terms of compression) produced by the SPCM with the SP-Patterns from Figure 18 in New and Old, as shown in that figure.

7.2 Ambiguities in language

Natural languages are notoriously ambiguous, not only in their meanings but also in their syntax. An example which includes syntactic ambiguity is the second sentence in 'Time flies like an arrow. Fruit flies like a banana', with uncertain origins.[14]

Figure 20 shows how the SPCM can accommodate the ambiguity of that example, given an appropriate grammar. In this example, the two parsings shown have compression values that are similar and higher than the compression scores of other alignments formed for the same sentence.

[14]Based on [1, Section 5.2.2].

```
0           fruit         flies                 like                a         banana       0
              |             |                    |                  |           |
1             |             |                    |                  |   < N 1 banana >     1
              |             |                    |                  |   | |          |
2             |             |                    |     < NP 1 < D   | > < N          > >   2
              |             |                    |     | |    | |   | |                |
3             |             |                    |     | |    < D 1 a >                |   3
              |             |                    |     | |                             |
4             |             |           < ADV 0 like > | |                             |   4
              |             |           |  |         | | |                             |
5             |             |     < ADP < ADV        > < NP                            > > 5
              |             |     |  |                                                   |
6     < N 2 fruit >         |     |  |                                                   | 6
      | |         |         |     |  |                                                   |
7 S 0 < N         > < V     |   > < ADP                                                  > 7
                    | |     |   |
8                   < V 0 flies >                                                          8

(a)

0                  fruit         flies           like                a         banana     0
                     |             |              |                  |           |
1                    |             |              |                  |   < N 1 banana >   1
                     |             |              |                  |   | |          |
2                    |             |              |     < NP 1 < D   | > < N          > > 2
                     |             |              |     | |    | |   | |                |
3                    |             |              |     | |    < D 1 a >                | 3
                     |             |              |     | |                             |
4                    |             |       < V 1 like > | |                             | 4
                     |             |       | |        | | |                             |
5 S 1 < NP           |             |     > < V        > < NP                            > 5
      | |            |             |     |
6     | |    < A 0 fruit >         |     |                                                6
      | |    | |         |         |     |
7     < NP 0 < A         > < N     |   > >                                                7
                           | |     |   |
8                          < N 0 flies >                                                  8

(b)
```

Figure 20: The two best SP-Multiple-Alignments found by the SPCM with SP-Patterns representing grammatical rules including words (in the repository of Old SP-patterbs) and the ambiguous sentence 'fruit flies like a banana' (which is a New SP-Pattern). Reproduced from [1, Figure 5.1].

7.3 Robustness in the face of errors of omission, addition, and substitution

Figure 21 (a) shows how the SPTI may achieve a 'correct' parsing of the sentence '`t w o k i t t e n s p l a y`', while Figure 21 (b) shows how the SPCM achieves the same 'correct' parsing, except that the sentence contains: an error of omission when the letter '`w`' is missing from the word '`t w o`'; an error of substitution when the letter '`m`' replaces the letter '`n`' in the word '`k i t t e n s`', and an error of addition when the letter '`x`' has been added to the word '`p l a y`'.[15]

[15]This section is based on [3, Section 4.2.2].

In effect, the parsing identifies errors in the sentence and suggests corrections for them: '`t o`' should be '`t w o`', '`k i t t e m s`' should be '`k i t t e n s`', and '`p l a x y`' should be '`p l a y`'.

```
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 > |                   | | | |       1
                             | | |          | |                | |                   | | | |
2                            | | |   < N Np < Nr               > s >                 | | | |       2
                             | | |   | | |                         |                 | | | |
3                   < D Dp 4 t w o > | | |                         |                 | | | |       3
                    | |            | | | |                         |                 | | | |
4              < NP < D            > < N |                         > >               | | | |       4
               | |                       |                           |               | | | |
5              | |                       |                           |        < Vr 1 p l a y >     5
               | |                       |                           |        | |           |
6              | |                       |                           | < V Vp < Vr        > >   6
               | |                       |                           | | | |                  |
7 < S Num    ; < NP                      |                           > < V |                  > > 7
       |     |                           |                                 |
8     Num PL ;                           Np                                Vp                      8

(a)

0                            t   o                 k i t t e     m s                   p l a x y       0
                             |   |                 | | | | |       |                   | | |   |
1                            |   |          < Nr 5 k i t t e n >   |                   | | |   |       1
                             |   |          | |                |   |                   | | |   |
2                            |   |   < N Np < Nr               >   s >                 | | |   |       2
                             |   |   | | |                           |                 | | |   |
3                   < D Dp 4 t w o > | | |                           |                 | | |   |       3
                    | |            | | | |                           |                 | | |   |
4              < NP < D            > < N |                           > >               | | |   |       4
               | |                       |                             |               | | |   |
5              | |                       |                             |        < Vr 1 p l a   y >     5
               | |                       |                             |        | |             |
6              | |                       |                             | < V Vp < Vr          > >   6
               | |                       |                             | | | |                    |
7 < S Num    ; < NP                      |                             > < V |                    > > 7
       |     |                           |                                   |
8     Num PL ;                           Np                                  Vp                        8

(b)
```

Figure 21: (a) The best SP-Multiple-Alignment created by the SP model with a New SP-Pattern (representing a sentence to be parsed) shown in row 0, and a store of Old SP-Patterns like those in rows 1 to 8 (representing grammatical structures, including words); and (b) The same as in (a) but with errors of omission, substitution and commission as described in the text, and with same set of Old SP-Patterns as before. (a) and (b) are reproduced from Figures 1 and 2 respectively in [47], with permission.

Examples like this suggest that—by contrast with the way in which DNNs are liable to make large and unexpected errors in recognition ([6, Section 3] and bullet point 2 in Section 8.2.1)—the SPTI and its realisation in the SPCM have robust capabilities for recovering from errors in its data.

7.4 Syntactic dependencies in French

As we have seen in Section 6.3.4, sentences in natural languages may contain syntactic dependencies between one part of a sentence and another.[16] As described in that section, there is usually a 'number' dependency between the subject of a sentence and the main verb of the sentence: if the subject has a *singular* form then the main verb must have a singular form and likewise for *plural* forms of the subject of a sentence and the main verb.

A prominent feature of these kinds of dependency is that they are often 'discontinuous' in the sense that the elements of the dependency can be separated, one from the next, by arbitrarily large amounts of intervening structure. For example, the subject and main verb of a sentence must have the same number (singular or plural) regardless of the size of qualifying phrases or subordinate clauses that may come between them.

Another interesting feature of syntactic dependencies, not discussed in Section 6.3.4, is that one kind of dependency, such as number dependency (*singular/plural*) can overlap other kinds of dependency, such as gender dependency (*masculine/feminine*), as can be seen in the following example.

In the French sentence *Les plumes sont vertes* ('The feathers are green') there are two sets of overlapping syntactic dependencies like this:

```
 P        P  P         P        Number dependencies
Les plume s sont vert e s
     F               F          Gender dependencies
```

In this example, there is a number dependency, which is plural ('P') in this case, between the subject of the sentence, the main verb and the following adjective: the subject is expressed with a plural determiner (*Les*) and a noun (*plume*) which is marked as plural with the suffix (*s*); the main verb (*sont*) has a plural form and the following adjective (*vert*) is marked as plural by the suffix (*s*).

Cutting right across these number dependencies is the gender dependency, which is feminine ('F') in this case, between the feminine noun (*plume*) and the adjective (*vert*) which has a feminine suffix (*e*).

For many years, linguists puzzled about how these kinds of syntactic dependency could be represented succinctly in grammars for natural languages. But then elegant solutions were found in transformational grammar [48] and, later, in systems like definite clause grammars [49], based on Prolog [50].

The solution proposed here is different from any established system and is arguably simpler and more transparent than other systems. It is described and illustrated here with a fragment of an SP-grammar of French, shown in Figure 22, which can generate the example sentence just shown.

Apart from the use of SP-Patterns as the medium of expression, this SP-grammar differs from systems like transformational grammar or definite clause grammars because the parts of the SP-grammar which express the forms of 'high level' structures like sentences, noun phrases and verb phrases (represented by the first four SP-Patterns in Figure 22) do not contain any reference to number or gender. Instead, the SP-grammar contains the eight SP-Patterns shown at the end of Figure 22.

In the SP-Multiple-Alignment shown in Figure 23, and in some that come later, SP-Multiple-Alignments are shown with SP-Patterns arranged in columns instead of rows (an option noted in Section 6.3.2). The choice of one arrangement or the other depends mainly on what fits best on

[16] Based on [1, Section 5.4].

```
S NP #NP VP #VP #S (500)
NP D #D N #N #NP (700)
VP 0 V #V A #A #VP (300)
VP 1 V #V P #P NP #NP #VP (200)
P 0 sur #P (50)
P 1 sous #P (150)
V SNG est #V (250)
V PL sont #V (250)
D SNG M 0 le #D (90)
D SNG M 1 un #D (120)
D SNG F 0 la #D (130)
D SNG F 1 une #D (110)
D PL 0 les #D (125)
D PL 1 des #D (125)
N NR #NR NS1 #NS1 #N (450)
NS1 SNG - #NS1 (250)
NS1 PL s #NS1 (200)
NR M papier #NR (300)
NR F plume #NR (400)
A AR #AR AS1 #AS1 AS2 #AS2 #A (300)
AS1 F e #AS1 (100)
AS1 M - #AS1 (200)
AS2 SNG - #AS2 (175)
AS2 PL s #AS2 (125)
AR 0 noir #AR (100)
AR 1 vert #AR (200)
NP SNG SNG #NP (450)
NP PL PL #NP (250)
NP M M #NP (450)
NP F F #NP (250)
N SNG V SNG A SNG (250)
N PL V PL A PL (250)
N M V A M (300)
N F V A F (400)
```

Figure 22: A fragment of French SP-grammar with SP-Patterns for number dependencies and gender dependencies—the last eight SP-Patterns in this SP-grammar.

the page. With SP-Multiple-Alignments in rows, New information is always in row 0 and Old information is in the remaining rows, one SP-Pattern in each row. When SP-Multiple-Alignments are in columns, New information is always in column 0 and Old information is always in the remaining columns, one SP-Pattern in each column.

The SP-Multiple-Alignment in the figure shows the best SP-Multiple-Alignment found by the SPCM with our example sentence in New and the SP-grammar from Figure 22 in Old. The main constituents of the sentence are marked in an appropriate manner and dependencies for number and gender are marked by SP-Patterns appearing in columns 13, 14 and 15 of the SP-Multiple-Alignment.

```
0       1      2      3      4      5     6      7      8       9     10    11    12     13    14  15

                                                                            S
                                                                      NP -- NP --------- NP
                                                                D --- D
                                                                PL --------------------- PL
                                                                0
les ----------------------------------------------------------- les
                                                                #D -- #D
                                                 N ------------------ N ---------------------- N - N
                                                 NR --- NR
                                                        F ------------------------------------ F
plume ------------------------------------------------- plume
                                                 #NR -- #NR
                                          NS1 -- NS1
                                          PL ------------------------------------------- PL ------ PL
s --------------------------------------- s
                                          #NS1 - #NS1
                                                 #N ----------------- #N
                                                                      #NP - #NP -------- #NP
                                    VP ------------------------------------ VP
                                    0
                             V ---- V -------------------------------------------------------- V - V
                             PL ------------------------------------------------------------------ PL
sont ----------------------- sont
                             #V --- #V
               A ------------------ A -------------------------------------------------------- A - A
               AR --- AR
                      1
vert ---------------- vert
               #AR -- #AR
        AS1 -- AS1
        F ------------------------------------------------------------------------------------ F
e ----- e
        #AS1 - #AS1
               AS2 -------------------------------------------------------------- AS2
                                                                                  PL ------------- PL
s ------------------------------------------------------------------------------- s
               #AS2 ------------------------------------------------------------- #AS2
               #A ----------------- #A
                                    #VP ----------------------------------- #VP
                                                                            #S

0       1      2      3      4      5     6      7      8       9     10    11    12     13    14  15
```

Figure 23: The best SP-Multiple-Alignment found by the SPCM with the New SP-Pattern in column 0, ‘`les plume s sont vert e s`’, representing a sentence, and Old SP-Patterns, one in each of columns 1 to 15 representing a grammatical structure or word, drawn from the SP-grammar shown in Figure 22. Reproduced from [1, Figure 5.8].

7.5 Dependencies in the syntax of english auxiliary verbs

This section presents an SP-grammar and examples showing how the syntax of English auxiliary verbs may be described in the SPTI.[17] Before the SP-grammar and examples are presented, the syntax of this part of English is described and alternative formalisms for describing the syntax are briefly discussed.

In English, the syntax for main verbs and the 'auxiliary' verbs which may accompany them follows two quasi-independent patterns of constraint which interact in an interesting way.

The *primary pattern of constraint* may be expressed with this sequence of symbols,

```
M H B B V,
```

which should be interpreted in the following way:

- Each letter represents a category for a single word:
  - '`M`' stands for 'modal' verbs like 'will', 'can', 'would', etc.
  - '`H`' stands for one of the various forms of the verb 'to have'.
  - Each of the two instances of '`B`' stands for one of the various forms of the verb 'to be'.
  - '`V`' stands for the main verb which can be any verb except a modal verb (unless the modal verb is used by itself).
- The words occur in the order shown but any of the words may be omitted.
- Questions of 'standard' form follow exactly the same pattern as statements except that the first verb, whatever it happens to be ('`M`', '`H`', the first '`B`', the second '`B`', or '`V`'), precedes the subject noun phrase instead of following it.

Here are two examples of the primary pattern with all of the words included:

```
It will have been being washed
    M    H    B     B     V

Will it have been being washed?
 M       H    B     B     V
```

The *secondary constraints* are these:

- Apart from the modals, which always have the same form, the first verb in the sequence, whatever it happens to be ('`H`', the first '`B`', the second '`B`' or '`V`'), always has a 'finite' form (the form it would take if it were used by itself with the subject).
- If an '`M`' auxiliary verb is chosen, then whatever follows it ('`H`', first '`B`', second '`B`', or '`V`') must have an 'infinitive' form (i.e., the 'standard' form of the verb as it occurs in the context 'to ...', but without the word 'to').

[17]This section is based on [1, Section 5.5].

- If an '`H`' auxiliary verb is chosen, then whatever follows it (the first '`B`', the second '`B`', or '`V`') must have a past tense form such as 'been', 'seen', 'gone', 'slept', 'wanted', etc. In Chomsky's *Syntactic Structures* [48], these forms were characterised as *en* forms and the same convention has been adopted here.

- If the first of the two '`B`' auxiliary verbs is chosen, then whatever follows it (the second '`B`' or '`V`') must have an *ing* form, e.g., 'singing', 'eating', 'having', 'being', etc.

- If the second of the two '`B`' auxiliary verbs is chosen, then whatever follows it (only the main verb is possible now) must have a past tense form (marked with *en* as above).

- The constraints apply to questions in exactly the same way as they do to statements.

Figure 24 shows a selection of examples with the dependencies marked.

```
           H------en  B2---------en
          ----    --  --         --
It  will  have  been  being  washed
    ----  ----  --      ---  ----
     M----inf   B1------ing   V
```

```
          B1------ing
          --      ---
Will  he  be  talking?
----      --  ----
 M-------inf   V
```

```
              V
            ------
They  have  finished
      ----        --
       H----------en
      fin
```

```
Are  they  gone?
---        ----
B2----------en
fin         V
```

```
         B1--------ing
         --        ---
Has  he  been  working?
---        --  ----
 H---------en   V
fin
```

Figure 24: A selection of example sentences in English with markings of dependencies between the verbs. *Key:* '`M`' = modal, '`H`' = forms of the verb 'have', '`B1`' = first instance of a form of the verb 'be', '`B2`' = second instance of a form of the verb 'be', '`V`' = main verb, '`fin`' = a finite form, '`inf`' = an infinitive form, '`en`' = a past tense form, '`ing`' = a verb ending in 'ing'.

7.5.1 Transformational grammar and english auxiliary verbs

In Figure 24 it can be seen that in many cases but not all, the dependencies which have been described may be regarded as discontinuous because they connect one word in the sequence to the suffix of the following word thus bridging the stem of the following word. Dependencies that are discontinuous can be seen most clearly in questions (e.g., the second, fourth and fifth sentences in Figure 24) where the verb before the subject influences the form of the verb that follows immediately after the subject.

In his book *Syntactic Structures*, [48], Noam Chomsky showed that this kind of regularity in the syntax of English auxiliary verbs could be described using transformational grammar. For each pair of symbols linked by a dependency (`M inf`, `H en`, `B1 ing`, `B2 en`) the two symbols could be shown together in the 'deep structure' of a sentence and then moved into their proper position or modified in form (or both) using 'transformational rules'.

This elegant demonstration argued persuasively in favour of transformational grammar compared with alternatives which were available at that time. However, as noted in Section 7.4, later research has shown that the same kinds of regularities in the syntax of English auxiliary verbs can be described quite well without recourse to transformational rules, using definite clause grammars or other systems which do not use that type of rule [49, 50]. An example showing how English auxiliary verbs may be described using the definite clause grammar formalism may be found in [51, pp. 183-184]).

7.5.2 English auxiliary verbs in the SPTI

Figure 25 shows an SP-grammar for English auxiliary verbs which exploits several of the ideas described above. Figures 26, 27 and 28 show the best SP-Multiple-Alignments in terms of information compression for three different sentences parsed by the SPCM model using the SP-grammar in Figure 25. In the following paragraphs, aspects of the SP-grammar and of the examples are described and discussed.

```
S ST NP #NP X1 #X1 XR #S (3000)
S Q X1 #X1 NP #NP XR #S (2000)
NP SNG it #NP (4000)
NP PL they #NP (1000)
X1 0 V M #V #X1 XR XH XB XB XV #S (1000)
X1 1 XH FIN #XH #X1 XR XB XB XV #S (900)
X1 2 XB1 FIN #XB1 #X1 XR XB XV #S (1900)
X1 3 V FIN #V #X1 XR #S (900)
XH V H #V #XH XB #S (200)
XB XB1 #XB1 XB #S (300)
XB XB1 #XB1 XV #S (300)
XB1 V B #V #XB1 (500)
XV V #V #S (5000)
M INF (2000)
H EN (2400)
B XB ING (2000)
B XV EN (700)
SNG SNG (2500)
PL PL (2500)
V M 0 will #V (2500)
V M 1 would #V (1000)
V M 2 could #V (500)
V H INF have #V (600)
V H PL FIN have #V (400)
V H SNG FIN has #V (200)
V H EN had #V (500)
V H FIN had #V (300)
V H ING hav ING1 #ING1 #V (400)
V B SNG FIN 0 is #V (500)
V B SNG FIN 1 was #V (400)
V B INF be #V (400)
V B EN be EN1 #EN1 #V (600)
V B ING be ING1 #ING1 #V (700)
V B PL FIN 0 are #V (300)
V B PL FIN 1 were #V (500)
V FIN wrote #V (166)
V INF 0 write #V (254)
V INF 1 chew #V (138)
V INF 2 walk #V (318)
V INF 3 wash #V (99)
V ING 0 chew ING1 #ING1 #V (623)
V ING 1 walk ING1 #ING1 #V (58)
V ING 2 wash ING1 #ING1 #V (102)
V EN 0 made #V (155)
V EN 1 brok EN1 #EN1 #V (254)
V EN 2 tak EN1 #EN1 #V (326)
V EN 3 lash ED #ED #V (160)
V EN 4 clasp ED #ED #V (635)
V EN 5 wash ED #ED #V (23)
ING1 ing #ING1 (1883)
EN1 en #EN1 (1180)
ED ed #ED (818)
```

Figure 25: An SP-grammar for the syntax of English auxiliary verbs.

7.5.3 The primary constraints

The first line in the SP-grammar is a sentence pattern for a statement (marked with the symbol '`ST`') and the second line is a sentence pattern for a question (marked with the symbol '`Q`'). Apart from these markers, the only difference between the two patterns is that, in the statement pattern, the symbols '`X1 #X1`' follow the noun phrase symbols ('`NP #NP`'), whereas in the question pattern they precede the noun phrase symbols. As can be seen in the examples in Figures 26, 27 and 28, the pair of symbols, '`X1 #X1`', has the effect of selecting the first verb in the sequence of auxiliary verbs and ensuring its correct position with respect to the noun phrase. In Figure 26 it follows the noun phrase, while in Figures 27 and 28 it precedes the noun phrase.

Each of the next four patterns in the SP-grammar have the form '`X1 ... #X1 XR ... #S`'. The symbols '`X1`' and '`#X1`' align with the same pair of symbols in the sentence pattern. The symbols '`XR ... #S`' encode the remainder of the sequence of verbs.

The first '`X1`' pattern encodes verb sequences which start with a modal verb ('`M`'), the second one is for verb sequences beginning with a finite form of the verb 'have' ('`H`'), the third is for sequences beginning with either of the two '`B`' verbs in the primary sequence (see below), and the last '`X1`' pattern is for sentences which contain a main verb without any auxiliaries.

In the first of the '`X1`' patterns, the subsequence '`XR ... #S`' encodes the remainder of the sequence of auxiliary verbs using the symbols '`XH XB XB XV`'. In a similar way, the subsequence '`XR ... #S`' within each of the other '`X1`' patterns encodes the verbs which follow the first verb in the sequence.

Notice that the pattern '`X1 2 XB1 FIN #XB1 #X1 XR XB XV #S`' can encode sentences which start with the first '`B`' verb and also contains the second '`B`' verb. And it also serves for any sentence which starts with the first or the second '`B`' verb with the omission of the other '`B`' verb. In the latter two cases, the 'slot' between the symbols '`XB`' and '`XV`' is left vacant. Figure 26 illustrates the case where the verb sequence starts with the first '`B`' verb with the omission of the second '`B`' verb. Figure 28 illustrates the case where the verb sequence starts with the second '`B`' verb (and the first '`B`' verb has been omitted).

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

                                                        S
                                                        ST
                                                  NP -- NP
                                                  SNG ------- SNG
it ---------------------------------------------- it
                                                  #NP - #NP
                                           X1 --------- X1
                                           2
                                    XB1 -- XB1
                              V --- V
                         B -- B --- B
                              SNG --------------------------- SNG
                              FIN -------- FIN
                              0
is -------------------------- is
                              #V -- #V
                                    #XB1 - #XB1
                                           #X1 -------- #X1
                                           XR --------- #XR
                                           XB
                    XV - XV -------------- XV
       V ---------- V
       EN -------------- EN
       5
wash - wash
       ED --- ED
ed ---------- ed
       #ED -- #ED
       #V --------- #V
                    #S ------------------- #S --------- #S

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

Figure 26: The best SP-Multiple-Alignment found by the SPCM with '`it is wash ed`' in New and the SP-grammar from Figure 25 in Old. Reproduced from [1, Figure 5.12].

```
0      1      2      3      4      5    6      7      8     9     10    11     12    13     14    15   16

                                                                  S
                                                                  Q
                                                      X1 -------- X1
                                                      O
                                               V ---- V
                                               M ---- M ----------------------------------- M
                                               O
will ----------------------------------------- will
                                               #V --- #V
                                                      #X1 ------- #X1
                                                            NP -- NP
                                                            SNG
it -------------------------------------------------------- it
                                                            #NP - #NP
                                                      XR -------- XR
                                                      XH --------------------- XH
                                                                        V ---- V
                                                                        H ---- H ---------------- H
                                                                        INF --------------- INF
have ------------------------------------------------------------------ have
                                                                        #V --- #V
                                                                               #XH
                                                      XB --------------------- XB
                                        XB ---------- XB
                     XB1 -------------- XB1
              V ---- V
              B ---- B ------------------------------------------------------------------------------- B
              EN -------------------------------------------------------------------------------- EN
be ---------- be
       EN1 -- EN1
en --- en
       #EN1 - #EN1
              #V --- #V
                     #XB1 ------------- #XB1
                                   XV - XV ---------- XV --------------------------------------------- XV
                            V ---- V
                            EN ----------------------------------------------------------------------- EN
                            1
brok ---------------------- brok
                            EN1 ---------------------------------------------------- EN1
en ---------------------------------------------------------------------------------- en
                            #EN1 ----------------------------------------------------#EN1
                            #V --- #V
                                   #S - #S ---------- #S -------- #S --------- #S

0      1      2      3      4      5    6      7      8     9     10    11     12    13     14    15   16
```

Figure 27: The best SP-Multiple-Alignment found by the SPCM with '`will it have be en brok en`' in New and the SP-grammar from Figure 25 in Old. Reproduced from [1, Figure 5.13].

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

                                   S
                                   Q
                                   X1 -- X1
                                         2
                                         XB1 -------- XB1
                                                V --- V
                                                B --- B --------- B
                                                PL --------- PL
                                         FIN -- FIN
                                                0
are ------------------------------------------- are
                                                #V -- #V
                                         #XB1 ------- #XB1
                                   #X1 - #X1
                            NP --- NP
                            PL ----------------------------- PL
they ---------------------- they
                            #NP -- #NP
                                   XR -- XR
                                         XB --------------------- XB
                       XV ---------------XV
       V --------------V
       ING ------------------------------------------------------ ING
       1
walk - walk
       ING1 -- ING1
ing ---------- ing
       #ING1 - #ING1
       #V ------------ #V
                       #S -------- #S -- #S

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

Figure 28: The best SP-Multiple-Alignment found by the SPCM with '`are they walk ing`' in New and the SP-grammar from Figure 25 in Old. Reproduced from [1, Figure 5.14].

7.5.4 The secondary constraints

The secondary constraints (Section 7.5) are represented using the patterns '`M INF`', '`H EN`', '`B XB ING`' and '`B XV EN`'. Singular and plural dependencies are marked in a similar way using the patterns '`SNG SNG`' and '`PL PL`'.

Examples appear in all three SP-Multiple-Alignments in Figures 26, 27 and 28. In every case except one (column 4 in Figure 26), the patterns representing secondary constraints appear in the later columns of the SP-Multiple-Alignment (towards the right). These examples show how dependencies bridging arbitrarily large amounts of structure, and dependencies that overlap each other, can be represented with simplicity and transparency in the medium of SP-Multiple-Alignments.

Notice, for example, how dependencies between the first and second verb in a sequence of auxiliary verbs are expressed in the same way regardless of whether the two verbs lie side by side (e.g., the statement in Figure 26) or whether they are separated from each other by the subject noun-phrase (e.g., the question in Figure 27 and in Figure 28). Notice, again, how the overlapping dependencies in Figure 27 and their independence from each other are expressed with simplicity and clarity in the SPTI.

7.6 The integration of syntax with semantics

In keeping with the remarks about the integration of diverse kinds of knowledge in Sections 5.1, 6.2, and 8.1.4, it has been anticipated that the SPTI would not only support the representation of syntactic and non-syntactic ('semantic') kinds of knowledge but that it would facilitate their integration.[18]

A preliminary example of how this might be done is shown in Figure 29. This is the best SP-Multiple-Alignment produced by the SPCM with 'john kissed mary' as the New pattern and a grammar in Old that contains patterns representing syntax, 'semantics' and the connections between them. The scare quotes are intended to indicate that the representations of semantic structures in this example are, at best, crude. That point made, the quote marks for 'semantics' or 'meanings' will be dropped in the remainder of this section.

In the SP-Multiple-Alignment, the sentence appears in column 0. In the remaining columns, Old patterns with the main roles are as follows:

- The pattern in column 2 represents the overall syntactic structure of the sentence: 'S AR NP #NP #AR AN V1 #V1 #AN OBJ NP #NP #OBJ #S'. Notice that this pattern differs from comparable patterns shown in previous examples because each constituent within the pattern is marked with its semantic role. Thus the first noun phrase ('NP #NP') is enclosed by the pair of symbols 'AR ... #AR' (representing the 'actor' role), the verb ('V1 #V1') is marked as an 'action' with the symbols 'AN ... #AN', and the second noun phrase is marked as an 'object' ('OBJ ... #OBJ').

- In column 7, the pattern 'A X an #an #X Y obj #obj #Y Z ar #ar #Z #A' may be seen as a generalised description of the association between an 'action' ('an #an'), the 'object' of the action ('obj #obj') and the 'actor' or performer of the action ('ar #ar'). Notice that the order in which these concepts are specified is different from the order of the corresponding markers in the syntax pattern. Notice also that these three slots are also marked, in order, as 'X ... #X', 'Y ... #Y' and 'Z ... #Z'. The reason for this marking will be seen shortly.

[18] This section is based on [1, Section 5.7].

```
0         1      2      3      4        5     6     7      8      9      10       11     12    13

                                              SM
                S -------------------------- S
                AR -------------------------------------------------------------------------- AR
                NP --- NP
                       1 -------------------------------------------------------------------- 1
john ----------------- john
                #NP -- #NP
                #AR ------------------------------------------------------------------------- #AR
                AN -------------------------------------------------------------------- AN
                V1 ------------------- V1
                              V ------ V
                              0 ----------------------------------------------------- 0
kissed ---------------------- kissed
                              #V ----- #V
                #V1 ------------------ #V1
                #AN ------------------------------------------------------------------- #AN
                OBJ -------------------------------------------- OBJ
         NP --- NP
         2 ----------------------------------------------------- 2
mary --- mary
         #NP -- #NP
                #OBJ ------------------------------------------- #OBJ
                #S ------------------------- #S
                                              A --- A
                                                    X --------------------------------- X
                                                    an ----------------- an
                                                                         0 ------------- 0
                                                                         KISSED
                                                    #an ---------------- #an
                                                    #X -------------------------------- #X
                                                    Y ----------- Y
                                                    obj -- obj
                                                           1 ---- 1
                                                           MARY
                                                    #obj - #obj
                                                    #Y ---------- #Y
                                                    Z ---------------------------------------- Z
                                                    ar -------------------------- ar
                                                                                  0 ---------- 0
                                                                                  JOHN
                                                    #ar ------------------------- #ar
                                                    #Z --------------------------------------- #Z
                                              #A -- #A
                                              #SM

0         1      2      3      4        5     6     7      8      9      10       11     12    13
```

Figure 29: The best SP-Multiple-Alignment created by the SPCM with the sentence 'john kissed mary' in New and a grammar in Old that represents natural language syntax, semantics and their integration. Reproduced from [1, Figure 5.18].

- In column 6, the pattern ‘SM S #S A #A #SM’ provides a link between the pattern in column 2 representing the syntactic structure of the sentence (‘S ... #S’) and the action-object-actor pattern in column 7 (‘A ... #A’).
- In columns 9, 12 and 13 are three more patterns that link the syntax with the semantics. In column 9, the pattern ‘OBJ 2 #OBJ Y 1 #Y’ connects ‘NP 2 mary #NP’ in the object position of the syntax with ‘MARY’ in the ‘obj #obj’ slot of the semantic structure. Here, ‘MARY’ is intended to represent some kind of conceptual structure that is the meaning of the word ‘mary’. More precisely, it is intended to represent a ‘code’ for that structure (see Section 7.6.1, below). In a similar way, the pattern in column 12 connects ‘kissed’ with ‘KISSED’ in the ‘an #an’ semantic slot; and the pattern in column 13 connects ‘john’ with ‘JOHN’ in the ‘ar #ar’ semantic slot.

The key idea in this example is that the SPTI allows information to be carried from the syntactic part of the knowledge structure to the semantic part and it allows the ordering of information to change from one part to the other. There seems no reason to suppose that this basic capability could not also be applied to examples in which the syntax and the semantics are more elaborate and more realistic.

7.6.1 Codes, meanings and the production of language from meanings

Section 8.3.3, describes how the SPTI can be used to produce a sentence, given a short code for that sentence supplied as New. That example shows in general terms how the system may be used for language production as well as language analysis but it seems unlikely that there would be many applications where there would be a requirement for the production of sentences purely in terms of their syntax and encodings of that syntax. In practice, it is more likely that one would wish to create sentences on the basis of intended *meanings*.

One possibility is that meanings might serve as codes for syntax and be used for language production in the way described in Section 8.3.3. In support of this view, we seem to remember what people have said in terms of meanings that were expressed rather than the words that were used to express them. And we can often reconstruct the words that people have used from the meanings that we remember—although there may be an element of lossy compression here because the reconstruction is not always accurate.

Figure 30 shows how, via the building of an SP-Multiple-Alignment, a sentence may be derived from the pattern ‘KISSED MARY JOHN’, representing an ‘internal’ code for the meaning to be expressed. The SP-Multiple-Alignment is the best SP-Multiple-Alignment created by the SPCM with that pattern in New and the same grammar in Old as was used for the example in Figure 29. In the top part of the SP-Multiple-Alignment, the words ‘john’, ‘kissed’, and ‘mary’ appear, in that order. If we strip out the ‘service’ symbols in the SP-Multiple-Alignment, we have the sentence corresponding to the semantic representation in column 0.

```
0        1        2      3      4      5     6      7      8     9     10       11    12     13

                                       SM
                                       S --- S
                                             AR ---------------- AR
                                             NP --- NP
                                                    1 ---------- 1
                                                    john
                                             #NP -- #NP
                                             #AR --------------- #AR
                                             AN ------------------------------- AN
                                             V1 ---------- V1
                                                           V --------- V
                                                                       0 ------ 0
                                                                       kissed
                                                           #V -------- #V
                                             #V1 --------- #V1
                                             #AN ------------------------------ #AN
                                             OBJ ------------------------------------------- OBJ
                                             NP ------------------------------------- NP
                                                                                      2 ---- 2
                                                                                      mary
                                             #NP ------------------------------------ #NP
                                             #OBJ ------------------------------------------ #OBJ
                                       #S -- #S
                  A ------------------ A
                  X ----------------------------------------------------------- X
         an ----- an
         0 -------------------------------------------------------------------- 0
KISSED - KISSED
         #an ---- #an
                  #X ---------------------------------------------------------- #X
                  Y ------------------------------------------------------------------------ Y
                  obj --------- obj
                                1 ---------------------------------------------------------- 1
MARY -------------------------- MARY
                  #obj -------- #obj
                  #Y ----------------------------------------------------------------------- #Y
                  Z -------------------------------------------- Z
                  ar --- ar
                         0 ------------------------------------- 0
JOHN ------------------- JOHN
                  #ar -- #ar
                  #Z ------------------------------------------- #Z
                  #A ----------------- #A
                                       #SM

0        1        2      3      4      5     6      7      8     9     10       11    12     13
```

Figure 30: The best SP-Multiple-Alignment created by the SPCM with ‘KISSED MARY JOHN’ in New and the same grammar in Old as was used for the example shown in Figure 29. Reproduced from [1, Figure 5.19].

7.7 Recognition and retrieval

In this section, Figure 31 provides an example of an SP-Multiple-Alignment created by the SPCM via a process of recognition, which may also be seen as a process of information retrieval.[19]

In this example, the caption to the figure tells us that this is the best SP-Multiple-Alignment found by the SPCM with the features of an unknown plant in column 0 and with SP-Patterns drawn from a repository of Old SP-Patterns like those shown in columns 1 to 6. This example shows how the SPCM may identify an unknown plant from its features. The answer of course is that the plant with the features shown in the SP-Multiple-Alignment is of the species 'acris' (Meadow Buttercup) (column 1), in the genus 'Ranunculus' (column 6), which is in the family 'Ranunculaceae' (column 5), and so on.

This process of recognition may also be regarded as a process of retrieval because it retrieves information that was not amongst the features of the unknown plant. We learn, for example, that each Meadow Buttercup photosynthesises (column 2), that the sepals are 'not reflexed' (column 1), that each flower has 5 petals (column 6), and so on.

Of particular interest in this example is that the SP-Multiple-Alignment illustrates how class-inclusion relations—for example, genus (column 1), family (column 6), order (column 5), and so on—and part-whole relations—for example, a shoot is compose of a stem, leaves, and flowers—may be combined in one SP-Multiple-Alignment.

[19]This section is based on [3, Section 10.1].

```
0                 1                2                  3              4              5                  6

                  <species>
                  acris
                  <genus> ---------------------------------------------------------------------------- <genus>
                  Ranunculus ------------------------------------------------------------------------- Ranunculus
                                                                                    <family> --------- <family>
                                                                                    Ranunculaceae ---- Ranunculaceae
                                                                     <order> ------ <order>
                                                                     Ranunculales - Ranunculales
                                                      <class> ------ <class>
                                                      Angiospermae - Angiospermae
                                   <phylum> --------- <phylum>
                                   Plants ----------- Plants
                                   <feeding>
has-chlorophyll ------------------ has-chlorophyll
                                   photosynthesises
                                   <feeding>
                                   <structure> ------ <structure>
                                                      <shoot>
<stem> ---------- <stem> ---------------------------- <stem>
hairy ----------- hairy
</stem> --------- </stem> --------------------------- </stem>
                  <leaves> -------------------------- <leaves>
                  compound
                  palmately-cut
                  </leaves> ------------------------- </leaves>
                                                      <flowers> ------------------- <flowers>
                                                                                    <arrangement>
                                                                                    regular
                                                                                    all-parts-free
                                                                                    </arrangement>
                  <sepals> -------------------------------------------------------- <sepals>
                  not-reflexed
                  </sepals> ------------------------------------------------------- </sepals>
<petals> -------- <petals> -------------------------------------------------------- <petals> --------- <petals>
                                                                                    <number> --------- <number>
                                                                                                       five
                                                                                    </number> -------- </number>
                  <colour> -------------------------------------------------------- <colour>
yellow ---------- yellow
                  </colour> ------------------------------------------------------- </colour>
</petals> ------- </petals> ------------------------------------------------------- </petals> -------- </petals>
                                                                                    <hermaphrodite>
<stamens> ------------------------------------------------------------------------- <stamens>
numerous -------------------------------------------------------------------------- numerous
</stamens> ------------------------------------------------------------------------ </stamens>
                                                                                    <pistil>
                                                                                    ovary
                                                                                    style
                                                                                    stigma
                                                                                    </pistil>
                                                                                    </hermaphrodite>
                                                      </flowers> ------------------ </flowers>
                                                      </shoot>
                                                      <root>
                                                      </root>
                                   </structure> ----- </structure>
<habitat> ------- <habitat> ------ <habitat>
meadows --------- meadows
</habitat> ------ </habitat> ----- </habitat>
                  <common-name> -- <common-name>
                  Meadow
                  Buttercup
                  </common-name> - </common-name>
                                   <food-value> ----------------------------------- <food-value>
                                                                                    poisonous
                                   </food-value> ---------------------------------- </food-value>
                                   </phylum> -------- </phylum>
                                                      </class> ----- </class>
                                                                     </order> ----- </order>
                                                                                    </family> -------- </family>
                  </genus> --------------------------------------------------------------------------- </genus>
                  </species>

0                 1                2                  3              4              5                  6
```

Figure 31: The best SP-Multiple-Alignment created by the SPCM, with a set of New SP-Patterns (in column 0) that describe some features of an unknown plant, and a set of Old SP-Patterns, including those shown in columns 1 to 6, that describe different categories of plant, with their parts and sub-parts, and other attributes. Reproduced from [3, Figure 16].

7.8 Medical diagnosis

```
<patient> John-Smith </patient>
<face> flushed </face>
<appetite> poor </appetite>
<breathing> rapid </breathing>
<muscles> aching </muscles>
<chills> yes </chills>
<fatigue> yes </fatigue>
<lymph-nodes> normal </lymph-nodes>
<malaise> no </malaise>
<nose> runny </nose>
<temperature> 38-39 </temperature>
<throat> sore </throat>
```

Figure 32: The set of New SP-Patterns supplied to the SPCM for the example discussed in the text. These patterns represent the patient ‘John Smith’ and his symptoms.

```
0                1                    2                    3              4                5

                 <disease> ---------- <disease> ---------- <disease> ---- <disease>
                                      flu
                 : ------------------ :
<patient> ------ <patient>
John-Smith
</patient> ----- </patient>
                 <dname> ------------ <dname>
                                      Influenza
                 </dname> ----------- </dname>
                 <R1> --------------- <R1>
                                      flu-symptoms ------- flu-symptoms
                 </R1> -------------- </R1>
                 <R2> ------------------------------------ <R2>
                                                           fever -------- fever
                 </R2> ----------------------------------- </R2>
<appetite> ----- <appetite> --------- <appetite>
poor                                  normal
</appetite> ---- </appetite> -------- </appetite>
<breathing> ---- <breathing> -------------------------------------------- <breathing>
rapid ------------------------------------------------------------------- rapid
</breathing> --- </breathing> ------------------------------------------- </breathing>
                 <chest> ------------ <chest>
                                      normal
                 </chest> ----------- </chest>
<chills> ------- <chills> -------------------------------- <chills>
yes ------------------------------------------------------ yes
</chills> ------ </chills> ------------------------------- </chills>
                 <cough> --------------------------------- <cough>
                                                           yes
                 </cough> -------------------------------- </cough>
                 <diarrhoea> -------- <diarrhoea>
                                      no
                 </diarrhoea> ------- </diarrhoea>
<face> --------- <face> ------------------------------------------------- <face>
flushed ----------------------------------------------------------------- flushed
</face> -------- </face> ------------------------------------------------ </face>
<fatigue> ------ <fatigue> ---------- <fatigue>
yes                                   no
</fatigue> ----- </fatigue> --------- </fatigue>
                 <headache> ------------------------------ <headache>
                                                           yes
                 </headache> ----------------------------- </headache>
<lymph-nodes> -- <lymph-nodes> ------ <lymph-nodes>
normal ------------------------------ normal
</lymph-nodes> - </lymph-nodes> ----- </lymph-nodes>
<malaise> ------ <malaise> ---------- <malaise>
no ---------------------------------- no
</malaise> ----- </malaise> --------- </malaise>
<muscles> ------ <muscles> ------------------------------- <muscles>
aching --------------------------------------------------- aching
</muscles> ----- </muscles> ------------------------------ </muscles>
<nose> --------- <nose> ---------------------------------- <nose>
runny ---------------------------------------------------- runny
</nose> -------- </nose> --------------------------------- </nose>
                 <skin> ------------- <skin>
                                      normal
                 </skin> ------------ </skin>
<temperature> -- <temperature> ------------------------------------------ <temperature>
                                                                          <t1> ----------- <t1>
38-39 ------------------------------------------------------------------------------------ 38-39
                                                                          </t1> ---------- </t1>
</temperature> - </temperature> ----------------------------------------- </temperature>
<throat> ------- <throat> -------------------------------- <throat>
sore ----------------------------------------------------- sore
</throat> ------ </throat> ------------------------------- </throat>
                 <weight-change> ---- <weight-change>
                                      no
                 </weight-change> --- </weight-change>
                 <causative-agent> -- <causative-agent>
                                      flu-virus
                 </causative-agent> - </causative-agent>
                 <treatment> -------- <treatment>
                                      flu-treatment
                 </treatment> ------- </treatment>
                 </disease> --------- </disease> --------- </disease> --- </disease>

0                1                    2                    3              4                5
```

Figure 33: The best alignment found by THE SPCM with the set of patterns from Figure 32 in New (describing the symptoms of the patient 'John Smith') and a set of patterns in Old describing a range of different diseases and named clusters of symptoms, together with the 'framework' pattern shown in column 1. Reproduced from [52, Figure 6].

7.9 Nonmonotonic reasoning and reasoning with default values

Conventional deductive reasoning is *monotonic* because deductions made on the strength of current knowledge cannot be invalidated by new knowledge: the conclusion that 'Socrates is mortal', deduced from 'All humans are mortal' and 'Socrates is human', remains true for all time, regardless of anything we learn later.[20] By contrast, the inference that 'Tweety can probably fly' from the propositions that 'Most birds fly' and 'Tweety is a bird' is *nonmonotonic* because it may be changed if, for example, we learn that Tweety is a penguin or an ostrich.

The elements of nonmonotonic reasoning are illustrated in the following SP-Multiple-Alignments.

```
0         1        2              3

                                  Default
                   Bd ----------- Bd
bird ------------- bird
          name --- name
Tweety - Tweety
          #name -- #name
                   f ------------ f
                                  canfly
                   #f ----------- #f
                   warm-blooded
                   wings
                   feathers
                   ...
                   #Bd ---------- #Bd
                                  #Default

0         1        2              3
```

Figure 34: The first of the three best SP-Multiple-Alignments formed by the SPCM with 'bird Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is calculated as 0.66. Reproduced from [3, Figure 17].

In Figure 34, a bird called 'Tweety' (columns 0 and 1) is identified as a bird (column 2) and, as such, it is assumed that it can (probably) fly (column 3). This inference is probabilistic because, as described below, there are two other SP-Multiple-Alignments that can be formed from the information that Tweety is a bird (columns 0 and 1). The SPTI calculates the relative probability of this SP-Multiple-Alignment as 0.66 (calculated from an imaginary frequency of occurrence assigned to each of the Old SP-Patterns).

In Figure 35, a bird called 'Tweety' (columns 0 and 1) is identified as a bird (column 2), and as an ostrich (column 3). In this case, we know that Tweety, as an ostrich, would not be able to fly (column 3), but because this SP-Multiple-Alignment is only one of three alternative SP-Multiple-Alignments created from the same New information, the result is less than certain. The relative probability of this SP-Multiple-Alignment is calculated as 0.22.

[20] This section is based on [3, Section 10.1].

```
0        1        2              3

                                 O
                                 ostrich
                  Bd ----------- Bd
bird ------------ bird
         name --- name
Tweety - Tweety
         #name -- #name
                  f ------------ f
                                 cannotfly
                  #f ----------- #f
                  warm-blooded
                  wings
                  feathers
                  ...
                  #Bd ---------- #Bd
                                 ...
                                 #O

0        1        2              3
```

Figure 35: The second of the three best SP-Multiple-Alignments formed by the SPCM with 'bird Tweety' in New and patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is calculated as 0.22. Reproduced from [3, Figure 18].

```
0        1        2               3

                                  P
                                  penguin
                  Bd ----------- Bd
bird ------------ bird
         name --- name
Tweety - Tweety
         #name -- #name
                  f ------------ f
                                  cannotfly
                  #f ----------- #f
                  warm-blooded
                  wings
                  feathers
                  ...
                  #Bd ---------- #Bd
                                  ...
                                  #P

0        1        2               3
```

Figure 36: The last of the three best SP-Multiple-Alignments formed by the SPCM with 'bird Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 0.12.

Figure 36, is much the same as Figure 35 except that, in this case, Tweety is a penguin and, as such, he (or she) would not be able to fly. The relative probability of this SP-Multiple-Alignment is calculated as 0.12.

```
0          1        2              3

                                   P
penguin ------------------------- penguin
                    Bd ----------- Bd
                    bird
          name --- name
Tweety -- Tweety
          #name -- #name
                    f ------------ f
                                   cannotfly
                    #f ----------- #f
                    warm-blooded
                    wings
                    feathers
                    ...
                    #Bd ---------- #Bd
                                   ...
                                   #P

0          1        2              3
```

Figure 37: The best SP-Multiple-Alignment formed by the SPCM with 'penguin Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 1.0.

Figure 37 is different from the previous three SP-Multiple-Alignments because, in this case, column 0 tells us that Tweety is a penguin, not a bird. Now there is a sharp change in the probability calculated by the SPTI: the relative probability calculated by the SPCM is 1.0 because there are no alternatives SMAS created from that New information in column 0. The same would apply if column 0 was 'ostrich Tweety'.

These examples illustrate nonmonotonic reasoning as outlined at the beginning of this section because the inferences that are made about Tweety and his (or her) ability to fly can change, depending on the information about Tweety that is supplied. This is much more in keeping with way people normally reason than is classical logic and its procrustean rules that prevents inferences from changing as we learn more about Tweety.

7.10 Explaining away 'explaining away': The SP Theory of Intelligence as an alternative to Bayesian networks

In recent years, *Bayesian networks* (otherwise known as *causal nets*, *influence diagrams*, *probabilistic networks* and other names) have become popular as a means of representing probabilistic knowledge and for probabilistic reasoning [53].

A Bayesian network is a directed, acyclic graph like the one shown in Figure 38 (below) where each node has zero or more 'inputs' (connections with nodes that can influence the given node) and one or more 'outputs' (connections to other nodes that the given node can influence).

Each node contains a set of conditional probability values, each one the probability of a given output value for a given input value or combination of input values. With this information, conditional probabilities of alternative outputs for any node may be computed for any given *combination* of inputs. By combining these calculations for sequences of nodes, probabilities may be propagated through the network from one or more 'start' nodes to one or more 'finishing' nodes.

This section shows how the SPTI provides an alternative to the Bayesian network explanation of the phenomenon of 'explaining away'.

7.10.1 A Bayesian network explanation of 'explaining away'

In [53, p. 7], Judea Pearl describes the phenomenon of 'explaining away' like this: 'If A implies B, C implies B, and B is true, then finding that C is true makes A *less* credible. In other words, finding a second explanation for an item of data makes the first explanation less credible.' (his italics). Here is an example:

> Normally an alarm sound alerts us to the possibility of a burglary. If somebody calls you at the office and tells you that your alarm went off, you will surely rush home in a hurry, even though there could be other causes for the alarm sound. If you hear a radio announcement that there was an earthquake nearby, and if the last false alarm you recall was triggered by an earthquake, then your certainty of a burglary will diminish. [53, pp. 8-9].

Although it is not normally presented as an example of nonmonotonic reasoning, this kind of effect in the way we react to new information is similar to the example we considered in Section 7.9 because new information has an impact on inferences that we formed on the basis of information that was available earlier.

The causal relationships in the example just described may be captured in a Bayesian network like the one shown in Figure 38.

Pearl argues that, with appropriate values for conditional probabilities, the phenomenon of 'explaining away' can be explained in terms of this network (representing the case where there is a radio announcement of an earthquake) compared with the same network without the node for 'radio announcement' (representing the situation where there is no radio announcement of an earthquake).

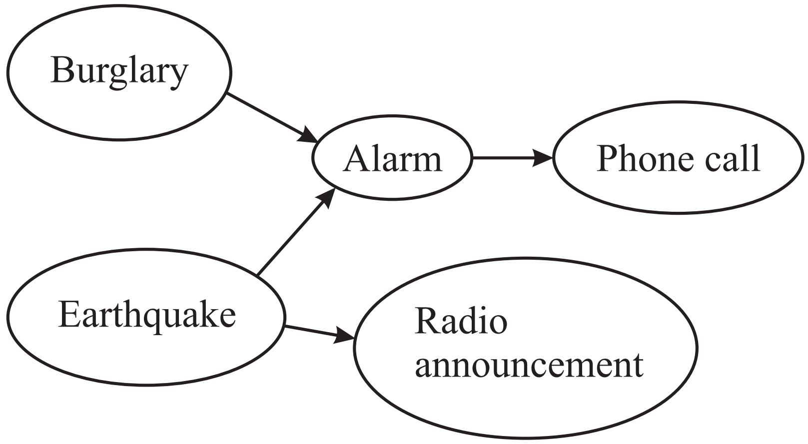

Figure 38: A Bayesian network representing causal relationships discussed in the text. In this diagram, 'Phone call' means 'a phone call about the alarm going off' and 'Radio announcement' means 'a radio announcement about an earthquake'.

7.10.2 Representing contingencies with SP-Patterns and frequencies

To see how this phenomenon may be understood in terms of the SP theory, consider, first, the set of SP-Patterns shown in Figure 39, which are to be stored in Old. The first four SP-Patterns in the figure show events which occur together in some notional sample of the 'World' together with their frequencies of occurrence in the sample.

Like other knowledge-based systems, an SPTI would normally be used with a 'closed-world' assumption that, for some particular domain, the knowledge stored in the knowledge base is comprehensive. Thus, for example, a travel booking clerk using a database of all flights between cities will assume that, if no flight is shown between, say, Edinburgh and Paris, then no such flight exists. Of course, the domain may be only 'flights provided by one particular airline', in which case the booking clerk would need to check databases for other airlines. In systems like Prolog, the closed-world assumption is the basis of 'negation as failure': if a proposition cannot be proved with the clauses provided in a Prolog program then, in terms of that store of knowledge, the proposition is assumed to be false.

In the present case, we shall assume that the closed-world assumption applies so that the absence of any SP-Pattern may be taken to mean that the corresponding SP-Pattern of events did not occur, at least not with a frequency greater than one would expect by chance.

The fourth SP-Pattern shows that there were 1000 occasions when there was a burglary and the alarm went off and the second SP-Pattern shows just 20 occasions when there was an earthquake and the alarm went off (presumably triggered by the earthquake). Thus we have assumed that, as triggers for the alarm, burglaries are much more common than earthquakes. Since there is no SP-Pattern showing the simultaneous occurrence of an earthquake, burglary and alarm, we shall infer from the closed-world assumption that this constellation of events was not recorded during the sampling period.

The first SP-Pattern shows that, out of the 1020 cases when the alarm went off, there were 980 cases where a telephone call about the alarm was made. Since there is no SP-Pattern showing

```
alarm phone-alarm-call (980)
earthquake alarm (20)
earthquake radio-earthquake-announcement (40)
burglary alarm (1000)
e1 earthquake e2 (40)
```

Figure 39: A set of SP-Patterns to be stored in Old in an example of 'explaining away'. The symbol 'phone-alarm-call' is intended to represent a phone call conveying news that the alarm sounded; 'radio-earthquake-announcement' represents an announcement on the radio that there has been an earthquake. The symbols 'e1' and 'e2' represent other contexts for 'earthquake' besides the contexts 'alarm' and 'radio-earthquake-announcement'.

telephone calls (about the alarm) in any other context, the closed-world assumption allows us to assume that there were no false positives (including hoaxes): telephone calls about the alarm when no alarm had sounded.

Some of the frequencies shown in Figure 39 are intended to reflect the two probabilities suggested for this example in [53, p. 49]: '... the [alarm] is sensitive to earthquakes and can be accidentally (P = 0.20) triggered by one. ... if an earthquake had occurred, it surely (P = 0.40) would be on the [radio] news.'

In our example, the frequency of 'earthquake alarm' is 20, the frequency of 'earthquake radio-earthquake-announcement' is 40 and the frequency of 'earthquake' in other contexts is 40. Since there is no SP-Pattern like 'earthquake alarm radio-earthquake-announcement' or 'earthquake radio-earthquake-announcement alarm' representing cases where an earthquake triggers the alarm and also leads to a radio announcement, we may assume that cases of that kind have not occurred. As before, this assumption is based on the closed-world assumption that the set of SP-Patterns is a reasonably comprehensive representation of non-random associations in this small world.

The SP-Pattern at the bottom, with its frequency, shows that an earthquake has occurred on 40 occasions in contexts where the alarm did not ring and there was no radio announcement.

7.10.3 Approximating the temporal order of events

In these SP-Patterns and in the SP-Multiple-Alignments shown below, the left-to-right order of symbols may be regarded as an approximation to the order of events in time. Thus, in the first SP-Pattern, 'phone-alarm-call' (a phone call to say the alarm has gone off) follows 'alarm' (the alarm itself); in the second SP-Pattern, 'alarm' follows 'earthquake' (the earthquake which, we may guess, triggered the alarm); and so on. A single dimension can only approximate the order of events in time because it cannot represent events which overlap in time or which occur simultaneously. However, this kind of approximation has little or no bearing on the points to be illustrated here.

7.10.4 Other considerations

Other points relating to the SP-Patterns shown in Figure 39 include:

- No attempt has been made to represent the idea that 'the last false alarm you recall was triggered by an earthquake' [53, p. 9]. At some stage in the development of the SPTI, there will be a need to take account of recency.

Symbol	*Probability*
alarm	1.0
burglary	0.3281
earthquake	0.0156

Table 1: The probabilities of unmatched symbols, calculated by the SPCM for the three SP-Multiple-Alignments shown in Figure 40.

- With these imaginary frequency values, it has been assumed that burglaries (with a total frequency of occurrence of 1160) are much more common than earthquakes (with a total frequency of 100). As we shall see, this difference reinforces the belief that there has been a burglary when it is known that the alarm has gone off (but without additional knowledge of an earthquake).
- In accordance with Pearl's example (p. 49) (but contrary to the phenomenon of looting during earthquakes), it has been assumed that earthquakes and burglaries are independent. If there was some association between them, then, in accordance with the closed-world assumption, there should be an SP-Pattern in Figure 39 representing the association.

7.10.5 Formation of SP-Multiple-Alignments: the burglar alarm has sounded

Receiving a phone call to say that one's house alarm has gone off may be represented by placing the symbol 'phone-alarm-call' in New. Figure 40 shows, at the top, the best SP-Multiple-Alignment formed by the SPCM in this case with the SP-Patterns from Figure 39 in Old. The other two SP-Multiple-Alignments in the reference set are shown below the best SP-Multiple-Alignment, in order of CD value and relative probability. The actual values for CD and relative probability are given in the caption to Figure 39.

The unmatched symbols in these SP-Multiple-Alignments represent inferences made by the system. The probabilities for these inferences which are calculated by the SPCM (using the method described in Section 6.6) are shown in Table 1. These probabilities do not add up to 1 and we should not expect them to because any given SP-Multiple-Alignment can contain two or more of these symbols.

The most probable inference is the rather trivial inference that the alarm has indeed sounded. This reflects the fact that there is no SP-Pattern in Figure 39 representing false positives for telephone calls about the alarm. Apart from the inference that the alarm has sounded, the most probable inference ($p = 0.3281$) is that there has been a burglary. However, there is a distinct possibility that there has been an earthquake—but the probability in this case ($p = 0.0156$) is much lower than the probability of a burglary.

These inferences and their relative probabilities seem to accord quite well with what one would naturally think following a telephone call to say that the burglar alarm at one's house has gone off (given that one was living in a part of the world where earthquakes were not vanishingly rare).

```
0       phone-alarm-call 0
               |
1 alarm phone-alarm-call 1

(a)

0                phone-alarm-call 0
                        |
1          alarm phone-alarm-call 1
             |
2 burglary alarm                  2

(b)

0                  phone-alarm-call 0
                          |
1            alarm phone-alarm-call 1
               |
2 earthquake alarm                  2

(c)
```

Figure 40: The best SP-Multiple-Alignment (at the top) and the other two SP-Multiple-Alignments in its reference set formed by the SPCM with the SP-Pattern 'phone-alarm-call' in New and the SP-Patterns from Figure 39 in Old. In order from the top, the values for CD with relative probabilities in brackets are: 19.91 (0.6563), 18.91 (0.3281), and 14.52 (0.0156).

7.10.6 Formation of SP-Multiple-Alignments: the burglar alarm has sounded and there is a radio announcement of an earthquake

In this example, the phenomenon of 'explaining away' occurs when you learn not only that the burglar alarm has sounded but that there has been an announcement on the radio that there has been an earthquake. In terms of the SP model, the two events (the phone call about the alarm and the announcement about the earthquake) can be represented in New by a SP-Pattern like this:

```
phone-alarm-call radio-earthquake-announcement
```

or 'radio-earthquake-announcement phone-alarm-call'. The order of the two symbols does not matter because it makes no difference to the result, except for the order in which columns appear in the best SP-Multiple-Alignment.

In this case, there is only one SP-Multiple-Alignment (shown at the top of Figure 41) that can 'explain' all the information in New. Since there is only this one SP-Multiple-Alignment in the reference set for the best SP-Multiple-Alignment, the associated probabilities of the inferences that can be read from the SP-Multiple-Alignment ('alarm' and 'earthquake') are 1.0: it was an earthquake that caused the alarm to go off (and led to the phone call) and not a burglary.

These results show broadly how 'explaining away' may be explained in terms of the SP theory. The main point is that the SP-Multiple-Alignment or SP-Multiple-Alignments that provide the best 'explanation' of a telephone call to say that one's burglar alarm has sounded is different from the SP-Multiple-Alignment or SP-Multiple-Alignments that best explain the same telephone call coupled with an announcement on the radio that there has been an earthquake. In the latter case, the best explanation is that the earthquake triggered the alarm. Other possible explanations have lower probabilities.

```
0                  phone-alarm-call radio-earthquake-announcement 0
                          |                        |
1            alarm phone-alarm-call                |              1
               |                                   |
2 earthquake alarm                                 |              2
      |                                            |
3 earthquake                        radio-earthquake-announcement 3
```

(a)

```
0 phone-alarm-call radio-earthquake-announcement 0
                                 |
1 earthquake       radio-earthquake-announcement 1
```

(b)

```
0       phone-alarm-call radio-earthquake-announcement 0
               |
1 alarm phone-alarm-call                               1
```

(c)

```
0                phone-alarm-call radio-earthquake-announcement 0
                        |
1          alarm phone-alarm-call                               1
             |
2 burglary alarm                                                2
```

(d)

```
0                  phone-alarm-call radio-earthquake-announcement 0
                          |
1            alarm phone-alarm-call                               1
               |
2 earthquake alarm                                                2
```

(e)

Figure 41: At the top, the best SP-Multiple-Alignment formed by the SPCM with the SP-Pattern 'phone-alarm-call radio-earthquake-announcement' in New and the SP-Patterns from Figure 39 in Old. Other SP-Multiple-Alignments formed by the SPCM are shown below. From the top, the CD values are: 74.64, 54.72, 19.92, 18.92, and 14.52.

7.10.7 Other possible SP-Multiple-Alignments

The foregoing account of 'explaining away' in terms of the SP theory is not entirely satisfactory because it does not say enough about alternative explanations of what has been observed. This subsection tries to plug this gap. What is missing from the account of 'explaining away' in the previous subsection is any consideration of such other possibilities as, for example:

- A burglary (which triggered the alarm) and, at the same time, an earthquake (which led to a radio announcement), or
- An earthquake that triggered the alarm and led to a radio announcement and, at the same time, a burglary that did not trigger the alarm.
- And many other unlikely possibilities of a similar kind.

Alternatives of this kind may be created by combining SP-Multiple-Alignments shown in Figure 41 with each other, or with SP-Patterns or symbols from Old, or both these things. The two examples just mentioned are shown in Figure 42.

```
0                phone-alarm-call radio-earthquake-announcement 0
                        |                        |
1          alarm phone-alarm-call                |              1
             |                                   |
2 burglary alarm                                 |              2
                                                 |
3 earthquake                      radio-earthquake-announcement 3

(a)

0                  phone-alarm-call radio-earthquake-announcement 0
                          |                        |
1            alarm phone-alarm-call                |              1
               |                                   |
2 earthquake alarm                                 |              2
      |                                            |
3 earthquake                        radio-earthquake-announcement 3

4 burglary                                                        4

(b)
```

Figure 42: Two SP-Multiple-Alignments discussed in the text. (a) An SP-Multiple-Alignment created by combining the second and fourth SP-Multiple-Alignment from Figure 41. $CD = 73.64$, Absolute P = 5.5391e-5. (b) An SP-Multiple-Alignment created from the first SP-Multiple-Alignment in Figure 41 and the symbol 'burglary'. $CD = 72.57$, Absolute P = 2.6384e-5.

Any SP-Multiple-Alignment created by combining SP-Multiple-Alignments as just described may be evaluated in exactly the same way as the SP-Multiple-Alignments formed directly by the SPCM. CDs and absolute probabilities for the two example SP-Multiple-Alignments are shown in the caption to Figure 42.

SP-Multiple-Alignment	*Absolute probability*	*Relative probability*
(a) in Figure 41	1.1052e-4	0.5775
(a) in Figure 42	5.5391e-5	0.2881
(b) in Figure 42	2.6384e-5	0.1372

Table 2: Values for absolute and relative probability for the best SP-Multiple-Alignment in Figure 41 and the two SP-Multiple-Alignments in Figure 42.

Given the existence of SP-Multiple-Alignments like those shown in Figure 42, values for relative probabilities of SP-Multiple-Alignments will change. The best SP-Multiple-Alignment from Figure 41 and the two SP-Multiple-Alignments from Figure 42 constitute a reference set because they all 'encode' the same symbols from New. However, there are probably several other SP-Multiple-Alignments that one could construct that would belong in the same reference set.

Given a reference set containing the first SP-Multiple-Alignment in Figure 41 and the two SP-Multiple-Alignments in Figure 42, values for relative probabilities are shown in Table 2, together with the absolute probabilities from which they were derived. Whichever measure is used, the SP-Multiple-Alignment which was originally judged to represent the best interpretation of the available facts has not been dislodged from this position.

7.10.8 The SP framework and Bayesian networks

The foregoing examples show that the SP framework is a viable alternative to Bayesian networks, at least in the kinds of situation that have been described. This subsection makes some general observations about the relative merits of the two frameworks for probabilistic reasoning where the events of interest are subject to multiple influences or chains of influence or both those things.

Bayes' theorem is a neat piece of mathematics and, as such, has much to commend it. But as the basis for theorising about the nature of science, knowledge, reasoning and so on, it has, in my view, certain shortcomings:

- *Simplicity in storing statistical knowledge.* As a medium for expressing statistical information, a Bayesian framework emphasises *conditional probabilities* rather than information about frequencies. Ultimately, the two ways of expressing statistical knowledge are equivalent but, in my view, a focus on frequencies cuts through many of the complexities that arise in trying to handle conditional probabilities. For example, each node in a Bayesian network contains a table of conditional probabilities for all possible combinations of inputs and these tables can be quite large. By contrast, the SP framework only requires a single measure of frequency for each SP-Pattern. The SP framework can calculate absolute probabilities or conditional probabilities as the need arises rather than storing its statistical knowledge in the form of conditional probabilities.

- *A focus on fundamentals.* By emphasising probabilities, Bayes' theorem is a distraction from simpler and more primitive ideas that underlie statistical concepts and seem to me to give a better handle on the issues. By contrast, the SP framework is focussed directly on primitive operations of matching and unification which, by hypothesis, provide the foundations for statistical abstractions. I believe this focus on fundamentals opens up possibilities that are closed when the primary focus is on higher-level concepts (see next point).

- *Creating ontologies from raw data.* Bayes' theorem assumes that the categories that are to be related to each other via conditional probabilities are already 'given' and so it is not very helpful in developing a theory which aims (amongst other things) to describe how ontological knowledge is created out of raw perceptual input. The SP framework gets round this difficulty by allowing new 'objects' and other categories to be derived from partial matches between SP-Patterns as outlined in Section 6.7.1.

7.11 Planning

Given New information about the desired start and finish of a traveller by air and a repository of Old information about direct flights between cities, the SPCM can work out alternative routes that may be taken. Five possibilities are shown in Figures 43 to 47

```
0           1          2            3

Beijing -- Beijing
           0
           Delhi --- Delhi
                     15a
                     Cape-Town - Cape-Town
                                 13
New-York ----------------------- New-York

0           1          2            3

(a)
```

Figure 43: This and the following four figures show SP-Multiple-Alignments representing a selection of routes between Beijing and New York. They are amongst the best SP-Multiple-Alignments found by the SPCM with 'Beijing New-York' in New and SP-Patterns showing one-way air links between individual cities as described in the text.

```
0          1         2       3

Beijing -- Beijing
           0
           Delhi --- Delhi
                     17
                     Paris - Paris
                             27
New-York ------------------- New-York

0          1         2       3

(b)
```

Figure 44: See Figure 43.

```
0          1           2           3

Beijing -- Beijing
           1
           Melbourne - Melbourne
                       22
                       Zurich ---- Zurich
                                   23a
New-York ------------------------- New-York

0          1           2           3

(c)
```

Figure 45: See Figure 43.

```
 0          1         2           3          4

 Beijing -- Beijing
            0
            Delhi --- Delhi
                      15a
                      Cape-Town ------------ Cape-Town
                                             14
                                  Paris ---- Paris
                                  25a
 New-York ---------------------- New-York

 0          1         2           3          4

 (d)
```

Figure 46: See Figure 43.

```
0          1          2           3           4

Beijing ------------------------- Beijing
                                  1
                                  Melbourne - Melbourne
                                              10a
                      Cape-Town ------------- Cape-Town
                      14
           Paris ---- Paris
           25a
New-York - New-York

0          1          2           3           4

(e)
```

Figure 47: See Figure 43.

7.12 Problem solving

As noted in Section 2.1.3, Barlow foresaw that '... the operations needed to find a less redundant code have a rather fascinating similarity to the task of answering an intelligence test, ...' [19, p. 210]. In support of his prescient observation, this section shows how IC at the core of the SPCM may solve a modified version of the kind of puzzle that is popular in intelligence tests.

Figure 48 shows an example of this type of puzzle. The task is to complete the relationship '`A is to B as C is to ?`' using one of the geometric SP-Patterns '`D`', '`E`', '`F`' or '`G`' in the position marked with '`?`' in the figure. For this example, the 'correct' answer is clearly '`E`'. Quote marks have been used for the word 'correct' because, in some problems of this type, there may be two or more alternative answers where there is uncertainty about which answer is the right one.

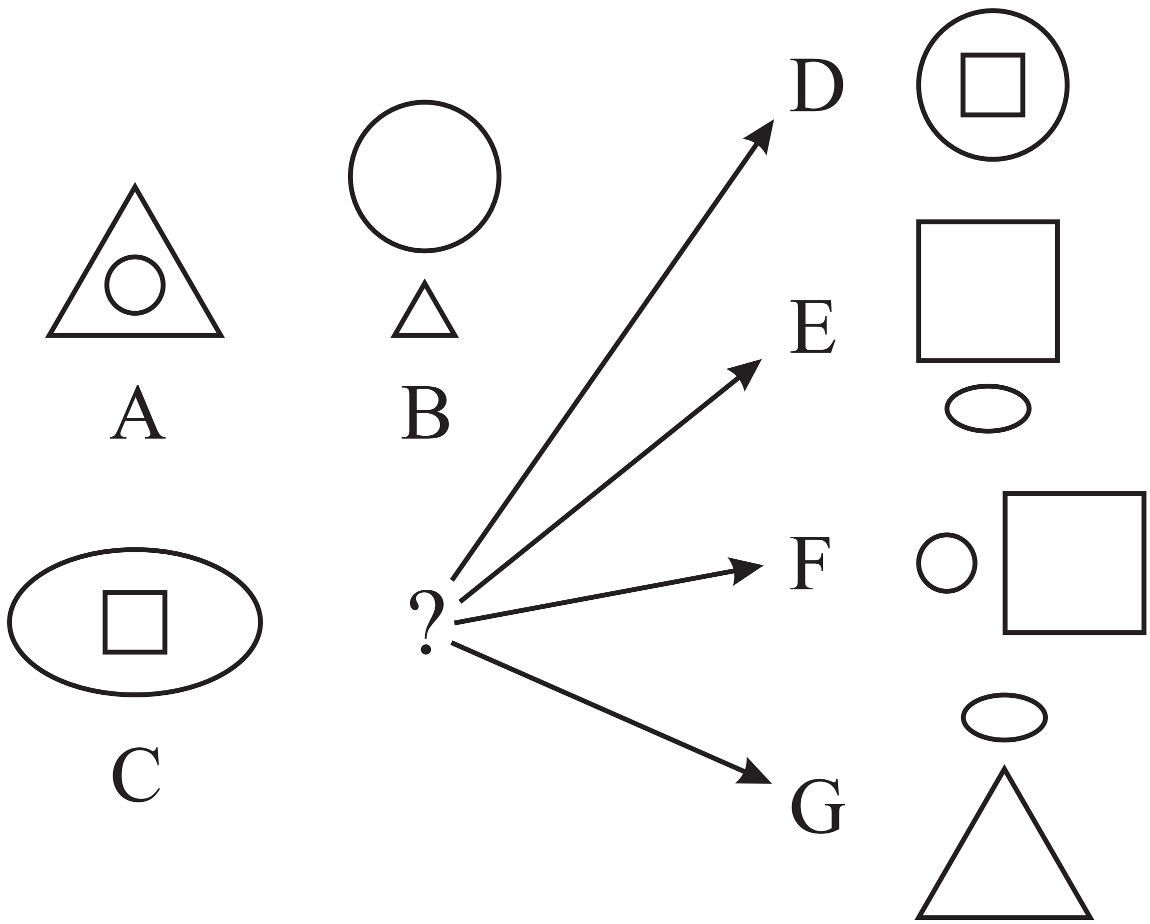

Figure 48: A geometric analogy problem.

Normally, these tests use simple geometric patterns like those shown in Figure 48 but because the SPCM has not yet been developed to process two-dimensional SP-Patterns, the geometric patterns are described textually as, for example, in '`small square inside large ellipse`', '`small square` `inside large circle`', and so on.

Computer-based methods for solving this kind of problem have existed for some time (e.g., Evans's [54] well-known heuristic algorithm). In more recent work [55, 56], AIT principles have been applied to good effect. The proposal here is that, within the general framework of Ockham's razor, this kind of problem may be understood in terms of the SP concepts.

Given that the diagrammatic form of the problem has been translated into textual patterns as described above, this kind of problem can be cast as a problem of partial matching, well within the scope of the SPCM.

Figure 49 shows the best SP-Multiple-Alignment created by the SPCM with New information in column 0 corresponding to the geometric patterns '`A`' and '`B`' in Figure 48, and the Old SP-Patterns shown in Figure 50 corresponding to the geometric patterns '`D`', '`E`', '`F`' and '`G`' in Figure 48.

```
0            1

             C2
small ---- small
circle     square
inside --- inside
large ---- large
triangle   ellipse
; -------- ;
             E
large ---- large
circle     square
above ---- above
small ---- small
triangle   ellipse
             #C2

0            1
```

Figure 49: The best SP-Multiple-Alignment found by the SPCM for the SP-Patterns in New and Old as described in the text.

As can be seen from the figure, the best SP-Multiple-Alignment found by the SPCM shows, in column 2, the combination of textual patterns corresponding to a combination of geometric patterns '`C`' and '`E`' in Figure 48, and of course this is the 'correct' answer as noted above.

As can be seen from the figure, finding the best SP-Multiple-Alignment from these New and Old SP-Patterns depends on the ability of the SPCM to find good partial matches between SP-Patterns.

```
C1 small square inside large ellipse ;
     D small square inside large circle #C1
C2 small square inside large ellipse ;
     E large square above small ellipse #C2
C3 small square inside large ellipse ;
     F small circle left-of large square #C3
C4 small square inside large ellipse ;
     G small ellipse above large rectangle #C4.
```

Figure 50: Textual patterns corresponding to the combination of 'C' on the bottom left of Figure 48 with one of 'D', 'E', 'F', or 'G' down the right side of the figure. These serve as Old SP-Patterns as described in the text.

8 The main strengths of the SP Theory of Intelligencen

In this book, the word 'strengths', applied to the SPTI, is intended to mean strengths that have been demonstrated with the SPCM and potential strengths that are more than mere speculations.[21]

Intelligence-related aspects of the SPTI are described quite fully in [3, Sections 5–12], and even more fully in [1, Chapters 5–9].

The first subsection here outlines the strength of the SPTI via the global concepts of Simplity and Power (Appendix B).

The two subsections that follow (Sections 8.1 and 8.2)) summarise the intelligence-related strengths of the SPTI, and the one after that (Section 8.3) summarises some other potential benefits and applications of the SPTI, less closely related to AI.

8.1 Aspects of intelligence exhibited by the SP Computer Model

The next four subsections describe intelligence-related features of the SPCM.

8.1.1 Intelligence in the SP Computer Model: excluding probabilistic reasoning)

The strengths of the SPTI in intelligence-related functions and other attributes are summarised in this section, excluding probabilistic reasoning which is given a separate subsection because of its importance.

- *Compression and decompression of information.* In view of substantial evidence for the importance of IC in human learning, peception, and cognition (Section 2), IC should be seen as an important feature of human intelligence.

- *Natural Language Processing.* Under the general heading of 'Natural Language Processing' are capabilities that facilitate the learning and use of natural languages. These include:

  - *Hierarchies of classes and sub-classes.* The ability to structure syntactic and semantic knowledge into hierarchies of classes and sub-classes, and into parts and sub-parts.

[21] This section is based on [36, Appendix A].

- *Integration of syntactic and semantic knowledge.* The ability to integrate syntactic and semantic knowledge. There is a simple example in Section 7.6.
- *Discontinuous dependencies in syntax.* The ability to encode discontinuous dependencies in syntax such as the number dependency (singular or plural) between the subject of a sentence and its main verb, or gender dependencies (masculine or feminine) in French—where 'discontinuous' means that the dependencies can jump over arbitrarily large intervening structures (Section 6.3.4, Figure 7).

  Also important in this connection is that different kinds of dependency (e.g., number and gender) can co-exist without interfering with each other (Section 7.4), and discontinuous dependencies provide an effective means of encoding the intricate structure of English auxiliary verbs (Section 7.5).
- *Representation of recursive structures in syntax.* The ability to accommodate recursive structures in syntax (see, for example,).
- *The production of natural language.* A point of interest here is that the SPCM provides for the production of language as well as the analysis of language, and it uses exactly the same processes for IC in the two cases—in the same way that the SPCM uses exactly the same processes for both the compression and decompression of information (Section 8.3.3).

• *Recognition and retrieval.* Capabilities that facilitate recognition of entities or retrieval of information include:

- *Recognition or retrieval via partial matches.* The ability to recognise something or retrieve information on the strength of a good partial match between features as well as an exact match.
- *Recognition or retrieval via classes and subclasses, and via parts and subparts.* Recognition or retrieval within a Class-Inclusion Hierarchy with 'inheritance' of attributes, and recognition or retrieval within an hierarchy of parts and sub-parts.
- *'Semantic' kinds of information retrieval.* 'Semantic' kinds of information retrieval—retrieving information via 'meanings'.
- *Computer vision.* Computer Vision [37], including visual learning of 3D structures (Section 6.7.6).

• *Planning and problem solving.* Capabilities here include:

- *Planning.* The ability to plan a route, such as for example a flying route between cities A and B, given information about direct flights between pairs of cities including those that may be assembled into a route between A and B.
- *Problem solving.* The ability to solve geometric analogy problems, or analogues in textual form.

• *Unsupervised Learning.* Chapter 9 of [1] describes how the SPCM may achieve unsupervised learning from a body of 'raw' data, **I**, to create an *SP-grammar*, **G**, and an **Encoding** of **I** in terms of **G**, where the encoding may be referred to as **E**. At present the learning process has shortcomings summarised in [3, Section 3.3] but it appears that these problems may be overcome.

In its essentials, unsupervised learning in the SPCM means searching for one or more 'good' SP-grammars, where a good SP-grammar is a set of SP-Patterns which is relatively effective in the economical encoding of **I** via an SP-Multiple-Alignment.

This kind of learning includes the discovery of segmental structures in data (including hierarchies of segments and subsegments) and the learning classes (including hierarchies of classes and subclasses).

8.1.2 Intelligence in the SP Computer Model: probabilistic reasoning

As described in [1, Chapter 7], several kinds of probabilistic reasoning flow from one relatively simple framework: the concept of SP-Multiple-Alignment:

- *One-step 'deductive' reasoning.* A simple example of *modus ponens* syllogistic reasoning goes like this:
  - If something is a bird then it can fly.
  - Tweety is a bird.
  - Therefore, Tweety can fly.
- *Abductive reasoning.* In brief, abductive reasoning means seeking the simplest and most likely conclusion from a set of observations, something which sits comfortably within the SPTI framework.
- *Probabilistic networks and trees.* One of the simplest kinds of system that supports reasoning in more than one step (as well as single step reasoning) is a 'decision network' or a 'decision tree'. In such a system, a path is traced through the network or tree from a start node to two or more alternative destination nodes depending on the answers to multiple-choice questions at intermediate nodes. Any such network or tree may be given a probabilistic dimension by attaching a value for probability or frequency to each of the alternative answers to questions at the intermediate nodes.
- *Reasoning with 'rules'.* SP-Patterns may serve very well within the SPCM for the expression of such probabilistic regularities as 'sunshine with broken glass may create fire', 'matches create fire', and the like. Alongside other information, rules like those may help determine one or more of the more likely scenarios leading to the burning down of a building, or a forest fire.
- *Nonmonotonic reasoning.* An example showing how the SPTI can perform nonmonotonic reasoning is described in Section 7.9. The conclusion that 'Socrates is mortal', deduced from 'All humans are mortal' and 'Socrates is human' remains true for all time, regardless of anything we learn later. By contrast, the inference that 'Tweety can probably fly' from the propositions that 'Most birds fly' and 'Tweety is a bird' is nonmonotonic because it may be changed if, for example, we learn that Tweety is a penguin. Section 7.9 shows how this works with SP-Multiple-Alignments.

- *'Explaining away'*. This means 'If A implies B, C implies B, and B is true, then finding that C is true makes A less credible.' In other words, finding a second explanation for an item of data makes the first explanation less credible.

There is also potential in the system for:

- *Spatial reasoning*. The potential is described in [40, Section IV-F.1].

- *What-if reasoning*. The potential is described in [40, Section IV-F.2].

8.1.3 Intelligence in the SP Computer Model: the representation and processing of several kinds of intelligence-related knowledge

Although SP-Patterns are not very expressive in themselves, they come to life in the SP-Multiple-Alignment framework within the SPCM. Within the SP-Multiple-Alignment framework, they provide relevant knowledge for each aspect of intelligence mentioned in Sections 8.1.1 and 8.1.2.

More specifically, they may serve in the representation and processing of such things as: 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 [47, Sections 3 and 4]; relational tuples [47, Sections 3] and concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition' ([1, Chapter 10], [57, Section 6.6.1], [58, Section 2]).

As noted in Section 6.2, the addition of two-dimensional SP-Patterns to the SPCM is likely to expand the capabilities of the SPTI to include the representation and processing of structures in two-dimensions and three-dimensions, and the representation of procedural knowledge with parallel processing.

8.1.4 Intelligence in the SP Computer Model: the seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination

An important additional feature of the SPCM, alongside its versatility in aspects of intelligence, including diverse forms of reasoning and its versatility in the representation and processing of diverse kinds of intelligence-related knowledge, is that *there is clear potential for the SPTI to provide for the seamless integration of diverse aspects of intelligence and diverse forms of knowledge, in any combination.*

This appears to be because those several aspects of intelligence, and several kinds of intelligence-related knowledge, all flow from a single coherent framework: SP-Patterns and SP-Symbols (Section 6.2), and the SP-Multiple-Alignment concept (Section 6.3).

It appears that this kind of seamless integration is *essential* in any artificial system that aspires to AGI.

Figure 51 shows schematically how the SPTI, with the SP-Multiple-Alignment at centre stage, exhibits versatility in diverse aspects of intelligence, and diverse kinds of intelligence-related knowledge, and their seamless integration.

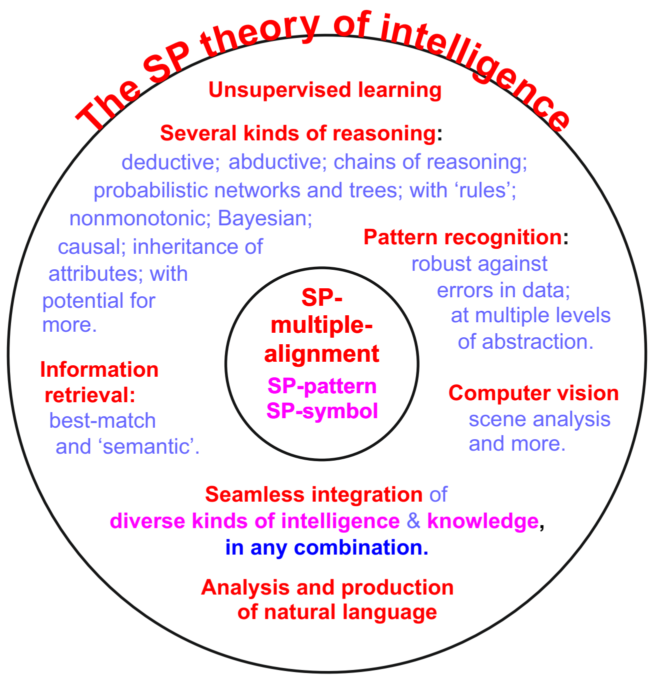

Figure 51: A schematic representation of versatility and seamless integration in the SPTI, with the SP-Multiple-Alignment concept, and the SP-Pattern and SP-Symbol concepts, centre stage.

8.2 Other aspects of intelligence relating to the SP Theory of Intelligence

This section describes some other evidence for the intelligence-related validity of the SPTI.

8.2.1 The clear potential of the SP Theory of Intelligence to solve twenty significant problems in artificial intelligence research

Strong evidence in support of the SPTI has arisen, indirectly, from the book *Architects of Intelligence* by science writer Martin Ford [34]. To prepare for the book, he interviewed several influential experts in AI to hear their views about AI research, including opportunities and problems in the field. He writes [34, p. 2]:

> The purpose of this book is to illuminate the field of artificial intelligence—as well as the opportunities and risks associated with it—by having a series of deep, wide-ranging conversations with some of the world's most prominent AI research scientists and entrepreneurs.

In the book, Ford reports what the AI experts say, giving them the opportunity to correct errors he may have made so that the text is a reliable description of their thinking.

This source of information has proved to be very useful in defining problems in AI research that influential experts in AI deem to be significant. This has been important from the SP perspective because, with 17 of those problems and three others—20 in all—there is clear potential for the SPTI to provide a solution.

Since these are problems with broad significance, not micro-problems of little consequence, the clear potential of the SPTI to solve them is a major result from the SP programme of research, demonstrating much of the power of the SPTI.

The peer-reviewed paper [6] describes those 20 significant problems, how the SPTI may solve them, with pointers to where fuller information may be found. *Because of the dominant position today of DNNs in AI research, most of these problems in that research are also problems with DNNs.*

The following summary describes each of the problems briefly with a summary of how the SPTI may solve it, together with an addition to point 14 below (about the importance of IC in intelligence). In addition, there is the same kind of summary for a twenty-first problem not mentioned in [6]:

1. *The Symbolic Versus Sub-Symbolic Divide.* The need to bridge the divide between symbolic and sub-symbolic kinds of knowledge and processing [6, Section 2]. The concept of an SP-Symbol can represent a relatively large symbolic kind of thing such as a word or a relatively fine-grained kind of thing such as a pixel.

2. *Errors in Recognition.* The tendency of DNNs to make large and unexpected errors in recognition [6, Section 3]. The overall workings of the SPTI and its ability to correct errors in data (Section 7.3) suggests that it is unlikely to suffer from these kinds of error.

3. *Natural Languages.* The need to strengthen the representation and processing of natural languages, including the understanding of natural languages and the production of natural language from meanings [6, Section 4]. The SPTI has potential in the representation and processing of several aspects of natural language.

4. *Unsupervised Learning.* Overcoming the challenges of unsupervised learning. Although DNNs can be used in unsupervised mode, they seem to lend themselves best to the supervised learning of tagged examples [6, Section 5]. Learning in the SPTI is entirely unsupervised.

   It is clear that most human learning, including the learning of our first language or languages [24], is achieved via unsupervised learning, without needing tagged examples, or reinforcement learning, or a 'teacher', or other form of assistance in learning (*cf.* [25]).

   Incidentally, a working hypothesis in the SP programme of research is that unsupervised learning can be the foundation for all other forms of learning, including learning by imitation, learning by being told, learning with rewards and punishments, and so on (Section 6.7).

5. *Generalisation, Over-Generalisation, and Under-Generalisation* (Section 6.7.4). The need for a coherent account of generalisation, over-generalisation (under-fitting) [6, Section 6], and under-generalisation (over-fitting).

   Although this is not mentioned in Ford's book [34], there is the related problem of reducing or eliminating the corrupting effect of errors in the data which is the basis of learning. The SPTI provides a coherent account of generalisation, and the correction of over- and under-generalisations, and avoiding the corrupting effect of errors in data.

6. *One-Shot Learning* (Section 8.2.1, [6, Section 7]). Unlike people, DNNs are ill-suited to the learning of usable knowledge from one exposure or experience. The ability to learn usable knowledge from a single exposure or experience is an integral and important part of the SPTI.

7. *Transfer Learning.* Although transfer learning—incorporating old learning in newer learning—can be done to some extent with DNNs [59, Section 2.1], DNNs fail to capture the fundamental importance of transfer learning for people, or the central importance of transfer learning in the SPCM [6, Section 8]. Transfer learning is an integral and important part of how the SPTI works.

8. *Reduced Demands for Data and Computational Resources Compared With DNNs* ([6, Section 9]). How to increase the speed of learning by DNNs, and how to reduce the demands of DNNs for large volumes of data, and for large computational resources [6, Section 9]. The ability of the SPTI to learn from a single exposure or experience, and the fact that transfer learning is an integral part of how it works, is likely to mean greatly reduced computational demands of the SPTI.

9. *Transparency.* The need for transparency in the representation and processing of knowledge [6, Section 10]. By contrast with DNNs, which are opaque in how they represent knowledge, and how they process it, the SPTI is entirely transparent in both the representation and processing of knowledge.

10. *Probabilistic Reasoning.* How to achieve probabilistic reasoning that integrates with other aspects of intelligence [6, Section 11]. The SPTI is entirely probabilistic in all its inferences, including the forms of reasoning described in [1, Chapter 7].

11. *Commonsense Reasoning an Commonsense Knowledge.* Unlike probabilistic reasoning, the area of commonsense reasoning and commonsense knowledge is surprisingly challenging [6, Section 12]. With qualifications, the SPTI shows some promise in this area [9, 60].

12. *How to Minimise the Risk of Accidents with Self-Driving Vehicles.* Notwithstanding the hype about self-driving vehicles, there are still significant problems in minimising the risk of accidents with such vehicles [6, Section 13]. The SPTI has potential in this area [61].

13. *Compositionality in the Representation of Knowledge.* DNNs are not well suited to the representation of Part-Whole Hierarchies or Class-Inclusion Hierarchies in knowledge [6, Section 14]. By contrast, the SPTI has robust capabilities in this area.

14. *Establishing the Importance of Information Compression in AI Research.* There is a need to spread the word about the significance of IC in both human learning, peception, and cognition and AI [6, Section 15]. The importance of IC in human learning, peception, and cognition is described in Section 2 and its importance in the SPTI is described in this and most other publications about the SPTI.

    A point which deserves emphasis which was not mentioned in [6] is that, while there is some recognition amongst other researchers of the importance of IC in machine learning, there appears to be little or no recognition of the importance of IC in other aspects of intelligence (see first bullet point in Section 1.3.3). The importance of IC in the SPTI across all aspects of intelligence, in keeping with evidence for the importance of IC in human learning, peception, and cognition (Section 2), is a major strength of the SPTI compared with other theories of intelligence.

15. *Establishing the Importance of a Biological Perspective in AI Research.* There is a need to raise awareness of the importance of a biological perspective in AI research [6, Section 16]. This is very much part of the SPTI research and publicity for that research.

16. *Distributed Versus Localist Representations for Knowledge.* A persistent issue in studies of human learning, peception, and cognition and in AI is whether knowledge in brains is represented in distributed or localist form, and which of those two forms works best in AI systems [6, Section 17]. DNNs employ a distributed form for knowledge, but the SPTI, which is firmly in the localist camp, has distinct advantages compared with DNNs. This reinforces other evidence for localist representations in brains.

17. *The Learning of Structures From Raw Data.* DNNs are weak in the learning of structures from raw data [6, Section 18]. By contrast, this is a clear advantage in the workings of the SPTI.

18. *The Need to Encourage Top-Down Strategies in AI Research.* Most AI research has adopted a bottom-up strategy, but this is failing to deliver generality in solutions [6, Section 19]. In the quest for AGI, there are clear advantages in adopting a top-down strategy (Appendix 3).

19. *Overcoming the Limited Scope For Adaptation in Deep Neural Networks.* An apparent problem with DNNs is that, unless many DNNs are joined together, each one is designed to learn only one concept, and the learning is restricted to what can be done with a fixed set of layers [6, Section 20]. By contrast, the SPTI, like people, can learn multiple concepts, and these multiple concepts are often in hierarchies of classes or in Part-Whole Hierarchies. This adaptability is largely because, via the SP-Multiple-Alignment concept, many different SP-Multiple-Alignments may be created in response to one body of data.

20. *The Problem of Catastrophic Forgetting.* Although there are somewhat clumsy workarounds for this problem, an ordinary DNN is prone to the problem of catastrophic forgetting, meaning that new learning wipes out old learning [6, Section 21]. There is no such problem with the SPTI which may store new learning independently with old learning, or form composite structures which preserve both old and new learning, in the manner of transfer learning (above).

21. *A Weakness of DNNs Not Mentioned in [6].* A matter which has become increasingly clear with further thought is that, despite the impressive things that have been done with DNNs,[22] DNNs are relatively restricted in the aspects of intelligence that, without augmentation, they can model. They show little of the versatility of the SPTI in modelling diverse aspects of intelligence (Sections 8.1 and 8.2), and in its other strengths (Section 8.3).

8.2.2 The SP Theory of Intelligence as a foundation for the development of artificial general intelligence

More evidence in support of the SPTI is presented in the peer-reviewed paper [9]. The paper argues that, since AGI is a long way from being achieved, we should assess AI projects and products as *foundations* for the development of AGI, not in terms of AGI itself. It is argued that, in those terms, the SPTI scores higher than AI products such as 'Gato' from DeepMind or 'DALLE 2' from OpenAI, largely because of the powerful SP-Multiple-Alignment concept and how it combines parsimony with intelligence-related versatility, and also because the central role for IC in the SPTI accords with the central role for IC in human learning, peception, and cognition (Section 2).

8.2.3 Commonsense reasoning and commonsense knowledge

An interesting aspect of AI is the challenging area of 'commonsense reasoning and commonsense knowledge', outlined under the fourth bullet point in Section 8.3.1 and described quite fully by Ernest Davis and Gary Marcus in [62].

Preliminary and unpublished papers about how the SPTI may be applied in this area of research may be downloaded via links from [63, 60].

8.3 Some other potential benefits and applications of the SP Theory of Intelligence

The three following subsections describe other potential benefits and applications of the SPTI.

8.3.1 A summary of papers about potential benefits and applications of the SP Theory of Intelligence

Here's a summary of peer-reviewed papers that have been published about potential benefits and applications of the SPTI:

1. *Overview.* The SPTI promises deeper insights and better solutions in several areas of application including: unsupervised learning, natural language processing, autonomous robots,

[22] Forming part of a system that has beaten the best human players at the game of Go, and forming part of a system that has automated the difficult task of working out likely 3D structures for sequences of amino-acid residues.

computer vision, intelligent databases, software engineering, information compression, medical diagnosis and big data. There is also potential in areas such as the semantic web, bioinformatics, structuring of documents, the detection of computer viruses, data fusion, new kinds of computer, and the development of scientific theories. The theory promises seamless integration of structures and functions within and between different areas of application [57].

2. *The Development of Intelligence in Autonomous Robots*. The SPTI opens up a radically new approach to the development of intelligence in autonomous robots [40].

3. *The Management of Big Data*. There is potential in the SPTI for the management of big data in the following areas: overcoming the problem of variety in big data; the unsupervised learning or discovery of 'natural' structures in big data; the interpretation of big data via pattern recognition, parsing and more; assimilating information as it is received, much as people do; making big data smaller; economies in the transmission of data; and more [7].

4. *Commonsense Reasoning and Commonsense Knowledge*. Largely because of research by Ernest Davis and Gary Marcus (see, for example, [62]), the challenges in this area of AI research are now better known. Preliminary work shows that the SPTI has promise in this area (Section 8.2.3).

5. *An Intelligent Database System*. The SPTI has potential in the development of an intelligent database system with several advantages compared with traditional database systems [47].

6. *Medical Diagnosis*. The SPTI may serve as a vehicle for medical knowledge and to assist practitioners in medical diagnosis, with potential for the automatic or semi-automatic learning of new knowledge [52].

7. *Natural Language Processing*. The SPTI has strengths in the processing of natural language ([3, Section 8], [1, Chapter 5]).

8. *Sustainability*. The SP Machine (Section 6.9.2), has the potential to reduce demands for energy from IT, especially in AI applications and in the processing of big data, in addition to reductions in CO_2 emissions when the energy comes from the burning of fossil fuels. There are several other possibilities for promoting sustainability [64].

9. *Transparency*. The SPTI with the SPCM provides an audit-trail for all its processing, and complete transparency in the way its output is structured [65].

10. *Vision, Both Artificial and Natural*. The SPTI opens up a new approach to the development of computer vision and its integration with other aspects of intelligence, and it throws light on several aspects of natural vision [37, 66].

8.3.2 The potential monetary value of the SP Theory of Intelligence and associated concepts

It seems that the potential monetary value of the SPTI and associated concepts could be substantial. Here is updated version of an argument in the Conclusion section of [57].

If, as a conservative estimate, the SPTI and associated concepts were to add 5% to the value of annual worldwide IT investments,[23] they would be worth $230 billion each year.

This figure assumes that the increase in value due to the SPTI and associated concepts would probably build up over a period of years but, thereafter, would be a continuing annual benefit into the future, in much the same way that the benefits of inventions like the telephone do not cease after one year but continue into the future. And the annual benefit would itself increase with the continuing increase of IT expenditure.

8.3.3 How one mechanism may achieve both the production and the analysis of data

An interesting feature of the SPCM is that SP-Multiple-Alignment processes for the analysis of New information are *exactly* the same as may be used for the production of information.

For example, with natural language, processes for the analysis of a sentence (Section 6.5.3) are, without any qualification, the same as may be used for the production of the same sentence (Section 6.5.4).

Since the SPCM works by compressing information, this feature of the SPCM looks, paradoxically, like 'decompression of information by compression of information' or more briefly 'decompression by compression'. How the whole system works, and how this paradox may be resolved, is explained in [3, Section 4.5] and [1, Section 3.8]. In brief:

- An initial compression of a body of information, **I**, may be achieved by the detection of repeating patterns in **I** and their merging or unification in the manner of ICMUP. With lossless IC, which is normally the case in this programme of research, each unified pattern would be given a relatively short identifier or code in accordance with the chunking-with-codes technique for IC.

  This kind of compression with encoding may be done through several different levels, as exemplified by the code SP-Pattern, '`S PL 0a 17 6 1 11 21 #S`', for the sentence, '`t h e p l u m s a r e r i p e`', described in Section 6.5.3.

- Provided there is some residual redundancy in the code SP-Symbols within the code pattern '`S PL 0a 17 6 1 11 21 #S`', the process may be run in reverse. This means that the code SP-Pattern is supplied to the SPCM as a New SP-Pattern, and the program is run again with the same set of Old SP-Patterns as when the code SP-Pattern was produced.

  The best SP-Multiple-Alignment produced by running the SPCM in reverse like this is shown in Figure 10. As noted in Section 6.5.4, all the words in the original sentence can be seen in the SP-Multiple-Alignment in the right order, thus recreating the whole sentence.

Although the recreation of the original sentence looks like decompression by compression, it actually achieved by precisely the same process as was used to achieve the original compression.

More generally, there is clear potential in the SPCM for the creation of entirely new structures which may be seen as novel or creative, but not necessarily artistic. This is an aspect of the SPTI that is waiting to be explored.

[23] Estimated to be $4.6 trillion in 2023 ('Worldwide IT spending is projected to total $4.6 trillion in 2023', Gartner Forecast, October 19, 2022, tinyurl.com/4dwn5kda., retrieved 2023-05-16.)

9 *SP-Neural*: a preliminary version of the SP Theory of Intelligence in terms of neurons and their inter-connections and inter-communications

The SPTI has been developed primarily in terms of abstract concepts such as the SP-Multiple-Alignment concept. However, a version of the SPTI called SP-Neural has also been described in outline, expressed in terms of neurons and their inter-connections and inter-communications ([10], [1, Chapter 11]).

It appears that, within SP-Neural, the SP concepts of SP-Pattern and SP-Multiple-Alignment can be expressed in terms of neurons and their interconnections. The main challenge is how the *processes* of building SP-Multiple-Alignments, and of unsupervised learning, can be expressed in terms of neural processes.

9.1 Neural inhibition

In view of evidence for the importance of neural *inhibition* in the workings of brains and nervous systems [67], and in view of evidence for the importance in IC in human learning, perception, and cognition [11], it seems possible that inhibition could be the neural basis for IC [10, Section 9].

What appears to be a promising line of attack is *the idea that inhibition plays the part of unification in the ICMUP concept of IC (Section 4)*:

- Unification in the ICMUP concept is when (within a body of information **I**) two or more patterns that match each other are reduced to a single instance.
- Providing that the patterns to be unified are more frequent within **I** than one would expect by chance, the merging of multiple instances to make one instance has the effect of removing redundancy from **I**.
- In a similar way, inhibition in the nervous system kicks in when two signals are the same. In lateral inhibition in the eye, for example, neighbouring fibres carrying incoming signals inhibit each other when they are both active [68, 69].

With regard to lateral inhibition, Larry Squire and colleagues write [70, p. 578]:

> Lateral inhibition represents the classic example of a general principle: most neurons in sensory systems are best adapted for detecting changes in the external environment. This principle can be explained in behavioural terms. As a rule, it is change that has the greatest significance for an animal—for example, the edge of an extended object or a static object beginning to move. This principle can also be explained in terms of information processing. Given a world that is filled with constants—with uniform objects, with objects that move only rarely—it is most efficient to respond only to changes.

Here, 'most efficient' may be read as 'contains least redundancy' or 'is most (losslessly) compressed'. Thus neural inhibition (lateral or otherwise) may be seen as the neural equivalent of IC in the SPTI, which, within that theory, is largely achieved via ICMUP, and especially via the SP-Multiple-Alignment concept.

As mentioned above, SP-Neural needs more development: much as with the important role of the SPCM in the development of the abstract version of the SPTI (Section 6.8), it seems likely that, in neuroscience as it is now, creating a computer model may be the most effective way of clarifying how neural inhibition may achieve IC, reducing vagueness in ideas, providing a means of testing ideas, and providing a means of demonstrating what can be done with the system when it is more mature.

9.2 Biological validity of SP-Neural

In the development of the SPTI the strategy has been to stay close to things that we know with some confidence (including aspects of our own human intelligence), and perhaps later to see whether the SPTI might have things to say about real neural structures and processes. A first step in that latter direction is *SP-Neural*, a 'neural' version of the SPTI (Section 1.2, the last bullet point but two, and 9).

Although SP-Neural is still at an early stage of development, it has potential to reflect the organisation of real neural networks more precisely than DNNs which are widely acknowledged to be only an approximate guess about the organisation of real neural networks.

10 Information compression provides an entirely novel perspective on the foundations of mathematics, logic, and computing

In view of evidence for the importance of IC in human learning, peception, and cognition (Section 2), and in view of the fact that mathematics is the product of human brains and has been designed as an aid to human thinking, it should not be surprising to find that IC is central in the structures and workings of mathematics.

This line of thinking is substantially different from any of the existing 'isms' in the foundations of mathematics, but there are weak connections with structuralism [8, Section 4.4.4].

The 'mathematics as IC' idea has things to say about the longstanding puzzle about why mathematics can be so effective in at least some parts of science (Section 10.7).

An important part of the arguments for the importance of IC in mathematics are the versions of ICMUP outlined in Section 4 and Section [8]. Examples are described in the following subsections.

10.1 Chunking-With-Codes in mathematics

The chunking-with-codes technique for IC (Section 4.2.2) is widely used in mathematics, as outlined in the following two subsections.

10.1.1 The basics

The basic idea is that a relatively large 'chunk' of information is given a relatively short identifier or 'code'. For example, with a function like $\sqrt{x}$, the symbol $\surd$ is the relatively short code and the chunk is the relatively large procedures for calculating square roots. In a similar way, with a function like $log(x)$, the word '`log`' is the relatively short code and the chunk is the relatively large procedures for calculating logarithms.

10.1.2 The number system as Chunking-With-Codes

The most primitive form of counting is simply to make a mark for each of several entities being counted—such as a mark on the wall for each day passing for a prisoner in his or her cell, a notch on his six-shooter by the bad guy in a western film for each man that he has killed, or a mark on paper for each of a group of sheep herded into a pen.

This kind of unary arithmetic works well with small numbers but is hopeless for bigger numbers, especially thousands or millions or more.

Chunking-with-codes solves this problem:

- A unary number like '0 1 1 1 1 1 1 1' is a relatively large chunk that can be given the relative short code, '7'. Likewise for numbers like '5', '9', and so on.

- A unary number like '0 1 1 1 1 1 1 1 1 1 1' is a relatively large chunk that can be given the relative short code, '10'. Here, the position of '1' in the second position to the left indicates that it represents the number of 10s, and '0' on the right incates the number of unary digits. Likewise for numbers like '20', '30', and so on. Here, the chunking-with-codes principle applies as it did with unary digits, but it applies to the number of 10s, not unary digits.

  In a similar way, two levels may be represented together in numbers like '28', '34', and so on.

- These principles may be applied in the same way with numbers like '200', '354', '622', and so on up to thousands, millions, and more.

10.2 Schema-Plus-Correction in mathematics

As outlined in Section 4.2.3, a 'schema' is a chunk that contains one or more 'corrections' to the chunk. Strictly speaking, the examples given for chunking-with-codes in Section 10.1 are examples of schema-plus-correction because the parameter, x, for each of $\sqrt{x}$ and $log(x)$, may be seen as a means of 'correcting' the schema by applying a different value for x on different occasions.

10.3 Run-length-coding in mathematics

As described in Appendix 4.2.4, run-length-coding is where some entity, pattern, or operation is repeated two or more times in an unbroken sequence. Then it may be reduced to a single instance with some indication that it repeats. In mathematics for example,

- *Addition.* An addition like $5+7$ may be seen as an example of the run-length coding technique for IC. In this case, $5 + 7$ may be seen as a compressed version of the procedure 'start with 5 and then: add 1, add 1, add 1, add 1, add 1, add 1, and add 1'. The seven applications of 'add 1' have been reduced to one.

- *Multiplication.* In a similar way, a multiplication like 3×8 may be seen as a compressed version of 'start with 0 and then: add 3, add 3, add 3, add 3, add 3, add 3, add 3, and add 3'.

- *The power notation.* The power notation, such as '10^9', is short for 10 with $\times 10$ repeated eight times, and is thus another example of run-length coding.

10.4 Combinations of these techniques

Further evidence for IC as a unifying principle in mathematics is in the combinations of the techniques like those described above in well-known equations and how they may achieve high levels of compression. For example:

- The equation $s = (gt^2)/2$, 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. It exhibits IC via run-length coding in multiplication and in the power notation.

- The equation $a^2 + b^2 = c^2$ is a very compact means of representing thousands or millions of different versions of Pythagoras's theorem applied to right-angled triangles. It exhibits IC via addition and via the power notation.

- Einstein's famous equation $e = mc^2$ is a very compact means of representing the relationship between energy (e), mass (m) and the speed of light (c) with thousands or millions of possible values for mass and corresponding values for energy. It exhibits IC via multiplication and via the power notation.

Other examples of ICMUP in mathematics are described in [8, Section 6.6].

10.5 Logic and computing

The kinds of arguments about the importance of IC in mathematics that are described in the preceding subsections may also be made about the importance of IC in logic and computing. Relevant arguments are described in [8, Section 7].

As noted in Section 1.2, for the sake of brevity, mathematics and logic will both be referred to as 'mathematics'. Computing is discussed in Section 13.

10.6 Mathematics, information compression, and probabilities

This and the following subsections described some issues relating to the 'Mathematics as IC' idea.

Since mathematics appears to be a set of techniques for IC and their application (as described in preceding parts of Section 10), and because of the close relation between IC and concepts of probability, described in Solomonoff's APT (Section 12), there is likely to be a probabilistic dimension to mathematics.

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—which is a key part of mathematics—has been shown to be fundamentally probabilistic. In that connection, Gregory Chaitin writes [71, p. 80]:

> 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.

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 SPTI, see Section 7.1), 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* [72].

Since it is likely that logic and computing may also be understood in terms of IC [8, Section 7], they may also have a probabilistic dimension.

10.7 Why is mathematics so unreasonably effective in the natural sciences?

As mentioned at the beginning of Section 10, this section provides a brief discussion, picking up on the description in [8, Introduction] of the often-expressed puzzlement about why mathematics can be so effective in science.

In an article called 'The unreasonable effectiveness of mathematics in the natural sciences', Eugene Wigner writes [73, p. 14]:

> The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve.

and similar things have been said by others.

However, Marcus Chown, quoting Stephen Wolfram, the creator of the symbolic computer language *Mathematica*, says that: [74, p. 265]:

> ... most of what is happening in the universe, such as the turbulence in the atmosphere and biology, is far too complex to be encapsulated by mathematical physics. ... we use mathematics to describe the only part of the universe that it is describable by mathematics.

It seems now that both of the points of view described above may be accommodated by the 'mathematics as IC' insight:

- Where there are informational redundancies in natural phenomena, such as the observation that, discounting the effects of air resistance and slight variations in the Earth's gravity in different places, the way objects of all weights accelerate as they fall, is the same everywhere on the Earth.

  The acceleration may be described by the formula $s = (gt^2)/2$ which may itself be understood in terms of the run-length coding technique for IC, both in multiplication and in the power notation (Section 10.4).

- But where there is little informational redundancy in natural phenomena—such as the haphazard motion of a leaf falling from a tree—it is difficult or impossible to achieve much IC with mathematics or anything else.

In connection with this topic, it is appropriate to mention that the concept of 'symmetry' has been invoked at least once (eg [38, pp. 18–19]) as an explanation for the unreasonable effectiveness of mathematics in science. But arguably symmetry is relatively complex (Section 4.3) and less well-defined compared with the ICMUP/SP-Multiple-Alignment explanation for the effectiveness of mathematics in (some parts of) science.

Notwithstanding the limitations of mathematics noted in Section 10.7, there are two main reasons for the unreasonable effectiveness of mathematics in science and in other areas:

- *Information compression.* In view of the arguments in Section 10, mathematics may be seen as a set of techniques for IC and their application.

- *Probabilities.* In view of pioneering work by Solomonoff (Aopendix D), there is an intimate connection between IC and concepts of probability (Appendix D).

11 A 'New Mathematics' as an integration of mathematics with the SP Theory of Intelligence

The idea that IC is central both in mathematics and in the SPTI suggests the creation of a *New Mathematics* (NM) as an integration of mathematics with mature versions of the SPTI. Drawing on [8, Section 9.2.1], there are potential benefits for many people, especially students, teachers, practitioners, and researchers in mathematics and science.

In brief, the potential benefits of the NM are:

- *Extending the range of applications of mathematics.* From the perspective of specialists in mathematics, the NM would greatly extend the range of potential applications of mathematics.

- *Bringing together two areas of strength.* The NM would benefit from more than two-thousand years of thinking about mathematics. At the same time, it will have all the strengths of the SPTI (Section 8), including strengths in several kinds of probabilistic reasoning (Section 8.1.2).

- *A potential synergy of mathematics with the SPTI.* The integration of mathematics and the SPTI may yield an NM which is more powerful than the two systems without integration. Probably, the NM would facilitate novel combinations of techniques bridging mathematics and the SPTI. In particular, mathematics that is supercharged with AI is likely to have a much greater range of potential applications than mathematics alone.

- *The integration of mathematics, logic, and computing.* Since 'logic' and 'computing' may be seen as a set of techniques for IC and their application, much as with mathematics [8, Section 7], there is potential for the NM to provide an integration of mathematics, logic, and computing.

- *New techniques for IC.* The NM is likely to open up both mathematics and science to new techniques for the succinct representation of knowledge, especially the powerful SP-Multiple-Alignment concept (Section 6.3).

- *The representation and processing of structures in two, three, and four dimensions.* There is potential to facilitate the learning, representation and processing of structures in two, three, and four dimensions (Section 6.7.6), where the fourth dimension is time as it features in videos and films.

- *Compatibility with how people think.* There is potential, via the SPTI, for the NM to provide everyone, especially researchers in mathematics and science, with methods for the representation and processing of knowledge that are more compatible with the way that people naturally think, to the extent that we understand those things.

- *Quantitative evaluation and comparison of scientific theories.* Using mathematics as a means of quantifying the *Simplicity* of any scientific theory, and its descriptive or explanatory *Power*, and thus facilitating quantitative comparisons amongst rival scientific theories.

- *Facilitating the integration of scientific theories.* There is potential to overcome some of the incompatibilities amongst scientific theories, including perhaps the longstanding problem of integrating quantum mechanics with relativity.

- *Creating scientific theories from data.* The automatic or semi-automatic creation of scientific theories from data [57, Section 6.10.7].

12 The potential of the SPTI as a theory of probabilities

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 new thinking: this section outlines some possibilities.

The subsections that follow expand on the potential of the SPTI with respect to different kinds of inferences and corresponding concepts of probability.

12.1 Inference and probabilities via generalisation

The kind of 'inductive' inference called 'generalisation' is part of unsupervised learning in the SPTI (Section 6.7.4.

12.2 Inferences and probabilities via partial matching in the SP-Multiple-Alignment concept

In the SPTI, partial matching within the SP-Multiple-Alignment framework is the basis for the making of many inferences [1, Section 7.2]:

> ... if a pattern is recognised from a subset of its parts (something that people and animals are very good at doing), then, in effect, an inference is made that the unseen part or parts are really there. We might, for example, recognise a car from seeing only the front half because the rear half is hidden behind another car or a building. The inference that the rear half is present is probabilistic because there is always a possibility that the rear half is absent or, in some surreal world, replaced by the front half of a horse, or something equally bizarre.
>
> In terms of [SP-Multiple-Alignments], [inferences] may be understood as the formation of [an SP-Multiple-Alignment] in which one or more [SP-Symbols] in the Old [SP-Patterns] are not aligned with any matching [SP-Symbol] or [SP-Symbols] in the New [SP-Pattern].

The strengths of the SPTI in the making of those kinds of inferences and the calculation of associated probabilities (Section 6.6) flow directly from the central role of ICMUP in the SP-Multiple-Alignment concept, and from the intimate relation between IC and concepts of probability (Appendix D).

More specifically, the SP-Multiple-Alignment concept within the SPTI has proved to be a powerful vehicle for several kinds of probabilistic reasoning ([3, Section 10], [1, Chapter 7]), and for their seamless integration in any combination (Section 8.1.4). Collectively, these several kinds of probabilistic reasoning, working together, have potential as a powerful aid to statistical inference.

12.3 Exploiting the asymmetry between information compression and concepts of probability

Solomonoff writes [13, p. 79]

> Both Huffman coding [[75]] and the 'information packing problem'[24] used probabilities to compress information. Algorithmic probability inverted this process and obtained probabilities from compression.

but there is nevertheless an asymmetry between IC and concepts of probability, as described in [8, Section 8.2]:

1. Absolute and conditional probabilities may be derived from SP-Multiple-Alignments within the SPTI (Section 6.6), but the patterns that match other within that SP-Multiple-Alignment may not be derived from probabilities.

2. Structures such as words and phrases (Section 2.4.1), and 2D, 3D, and 4D structures (Section 6.7.6), and corresponding probabilities, may be derived via IC, but that kind of potential appears to be missing from all kinds of analysis of probabilities.

3. As described in [8, Section 8.2.4], it is not possible to derive causations from probabilities, but it is possible to derive causations from structures created via ICMUP as outlined in point 2 above.

These asymmetries mean that there are likely to be advantages in working from IC to probabilities, but not the other way round.

[24] "... how much data could one pack into a fixed number of bits, or conversely, how could one store a certain body of data using the least number of bits?", [13, p. 75]

12.4 Towards a new science of probability

The ideas described in preceding subsections have potential as the basis for a new science of probability:

- *Statistical analysis via unsupervised learning.* It appears that, because of the intimate relation between IC and probabilities (Appendix D), compression of a body of data via unsupervised learning in the SPTI is, in effect, a comprehensive statistical analysis of those data.

- *Making good use of small frequencies.* It is often assumed that, when the frequency of occurrence of entities or events is used as the basis of probability measures, high frequencies are needed to ensure that results are statistically significant.[25] But with ICMUP, as explained in [8, Section 8.2.3], the sizes of repeating patterns are as important as their frequency—which means that with matches between medium-to-large patterns, frequencies as low as 1 or 2 can be statistically significant.

  In case this seems to break all the rules of probability and statistics, consider how one can, often with a high degree of confidence, identify a song from hearing only one or two short extracts from the song. In a similar way, "It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife." is quite sufficient to identify Jane Austen's *Pride and Prejudice*. In both these examples, there is a partial match between some kind of search pattern and what is typically a much larger pattern stored in one's memory.

- *Modelling bayesian networks via the SPTI.* The SPTI has proved to be an effective alternative to Bayesian reasoning, including reasoning in Bayesian networks ([3, Section 10.2], [1, Section 7.8]).

13 The potential of the SPTI as a theory of computing

This section considers the potential of the SPTI as a model of 'computing', starting with an outline of the structure and workings of the Post Canonical System (PCS) as a basis for arguments that come later.

Later in this section, we consider some of the potential benefits of the SPTI as a model of computing.

13.1 Concepts of computing and the PCS

The nature of 'computing' was the focus of much interest in the 1930s and early '40s but, since then, it has been widely accepted that the essentials of this concept have been captured in Alan Turing's 'Universal Turing Machine' [76] and that other models of computing (such as 'Lambda Calculus' (Church and Kleene, see [77]), 'Recursive Function' [78], 'Normal Algorithm' [79] and Post's 'Canonical System' [80] are equivalent.

[25] See, for example, "There is a definition of probability in terms of frequency that is sometimes usable. It tells us that a good estimate of the probability of an event is the frequency with which it has occurred in the past. This simple definition is fine in many situations, but breaks down when we need it most; i.e., its precision decreases markedly as the [sample size] decreases. For sample sizes of 1 or 2 or none, the method is essentially useless." [13, pp. 74–75].

These concepts of computing have been monumentally successful and provide the theoretical underpinnings for much of the extraordinary development of digital computers up to now. However, although Alan Turing saw that computers might become intelligent [81], the Turing model, in itself, does not tell us how. If it did, then all of the research in artificial intelligence over the last 70 or so years would not have been necessary.

As noted in the Introduction (Section 1) and in Section 8.2.2, the SPTI should best be regarded as a foundation for the development of AGI, not yet a comprehensive theory of AGI, or close to that. The suggestion here is that, in addition to it's potential for the development of AGI, the SPTI has potential to be developed into a general model of computing.

As we have seen, the SPTI, as it is realised in the SPCM, performs all its computing by compressing information (Sections 6, 6.3, and 6.7). The suggestion here is that, in principle, anything that may be computed with a universal Turing machine—which is widely accepted as a comprehensive definition of computing—may also be computed by some relatively mature version of the SPCM.

Evidently, the 'in principle' and 'relatively mature' qualifications in the previous paragraph mean that this proposal is not yet rock solid. But there is evidence for the idea, considered in what follows.

13.1.1 The structure and workings of a PCS

This subsection describes the PCS briefly, since later arguments depend on an understanding of the PCS.

A PCS comprises:

- An *alphabet* of primitive *symbols* ('letters' in Post's terminology),
- One or more *primitive assertions* or *axioms*. These can often be regarded as 'input'.
- One or more *productions* which can often be regarded as a 'program'.

Each production has this general form:

$$g\_0 \ \$\_1 \ g\_1 \ \$\_2 \ ... \ \$\_n \ g\_n \rightarrow h\_0 \ \$'\_1 \ h\_1 \ \$'\_2 \ ... \ \$'\_m \ h\_m$$

where Each g_i and h_i is a certain fixed string; g_0 and g_n are often null, and some of the h's can be null.[26] Each $\$\_i$ is an 'arbitrary' or 'variable' string, which can be null. Each $\$'\_i$ is to be replaced by a certain one of the $\$\_i$.' [82, pp. 230–231].

In its simplest 'normal' form, a PCS has one primitive assertion and each production has the form:

$$g \ \$ \rightarrow \$ \ h$$

where g and h each represent a string of zero or more symbols, and both instances of '\$' represent a single 'variable' which may have a 'value' comprising a string of zero or more symbols.

It has been proved [80] that any kind of PCS can be reduced to a PCS in normal form [82, Chapter 13]. That being so, a PCS in this form will be the main focus of our attention.

[26] Spaces between symbols here and in other examples have been inserted for the sake of readability and because it allows us to use atomic symbols where each one comprises a string of two or more non-space characters (with spaces showing the start and finish of each string). Otherwise, spaces may be ignored.

13.1.2 How the PCS works

When a PCS (in normal form) processes an 'input' string, the first step is to find a match between that string and the left-hand side of one of the productions in the given set. The input string matches the left hand side of a production if a match can be found between leading symbols of the input string and the fixed string (if any) at the start of that left-hand side, with the assignment of any trailing substring within the input string to the variable within the left-hand side of the production.

Consider, for example, a PCS comprising the alphabet 'a ... z', an axiom or input string 'a b c b t', and productions in normal form like this:

$$\begin{array}{l} \text{a \$} \rightarrow \text{\$ a} \\ \text{b \$} \rightarrow \text{\$ b} \\ \text{c \$} \rightarrow \text{\$ c.} \end{array}$$

In this example, the first symbol of the input string matches the first symbol in the first production, while the trailing 'b c b t' is understood to match the variable and to become the value of that variable. The result of a successful match like this is that a new string is created in accordance with the configuration on the right hand side of the production which has been matched. In the example, the new string would have the form 'b c b t a', derived from 'b c b t' in the variable and 'a' which follows the variable on the right hand side of the production.

After the first step, the new string is treated as new input which is processed in exactly the same way as before. In this example, the first symbol of 'b c d t a' matches the first symbol of the second production, the variable in that production takes 'c d t a' as its value and the result is the string 'c b t a b'.

This cycle is repeated until matching fails. It should be clear from this example that the effect would be to 'rotate' the original string until it has the form 't a b c b'. The 't' which was at the end of the string when processing started has been brought round to the front of the string—and causes the process to stop because 't' does not match any of the characters in the left sides of any of the productions.

This is an example of the 'rotation trick' used by [82, Chapter 13] in demonstrating how a PCS in normal form can model any kind of PCS.

With some combinations of alphabet, input and productions, the process of matching strings to productions never terminates. With some combinations of alphabet, input and productions, the system may follow two or more 'paths' to two or more different 'conclusions' or may reach a given conclusion by two or more different routes. The 'output' of the computation is the set of strings created as the process proceeds.

13.1.3 With the PCS, the creation and recognition of numbers in unary notation

In the unary number system, 0 = 0, 1 = 01, 2 = 011, 3 = 0111, and so on. The unary number system can be defined with a PCS like this:

- Alphabet: the symbols 0 and 1.
- Axiom: 0.
- Production: If any string '\$' is a number, then so is the string '\$ 1'.

New

0 (Other options for New are described in the text)

2 *Old*

X a 0 #X
X b X #X 1 #X

Figure 52: SP-Patterns corresponding to a PCS for the creation or recognition of unary numbers.

This can be expressed with the production:

$$\$ \rightarrow \$\ 1$$

Since if x is a unary number then x followed by 1 is a unary number, this PCS is recursive and can be used to create the infinite series of unary strings: '0', '0 1', '0 1 1', '0 1 1 1', '0 1 1 1 1' etc, as far as resources allow.

Slightly less obviously, the PCS can also be used to recognise a string of symbols as being an example of a unary number. This is done by using the production in 'reverse', matching a character string to the right hand side of the production, taking the left hand side as the 'output' and then repeating the right-to-left process until only the axiom will match.

13.1.4 With the SP Computer Model, the creation and recognition of numbers in unary notation

Figure 52 shows a New SP-Pattern and two Old SP-Patterns to model the example of a PCS for the creation or recognition of unary numbers (Section 13.1.3, above).

The New SP-Pattern corresponds to the axiom in the PCS while the pattern '`X b X #X 1 #X`' is equivalent to the production. The pattern '`X a 0 #X`' represents the number 0 corresponding to 0 in the alphabet of the PCS. Incidentally, '`a`' and '`b`' in the two Old SP-Patterns are needed merely to accommodate the scoring system in the SPCM and may otherwise be ignored.

The pair of symbols '`X #X`' in the pattern '`X b X #X 1 #X`' may be read as 'any unary number' and the whole pattern may be read as 'a unary number is any unary number followed by 1', a recursive description much like the production '$\$ \rightarrow \$\ 1$'.

Given the pattern '`0`' in New and the other two patterns in Old, the SPCM creates a succession of good SP-Multiple-Alignments, one example of which is shown in Figure 53 (a). If it is not stopped, the program will continue producing SP-Multiple-Alignments like this until the memory of the machine is exhausted.

If we project the SP-Multiple-Alignment in Figure 53 (a) into a single sequence and then ignore the 'service' symbols ('`a`', '`b`', '`X`' and '`#X`'), we can see that the system has, in effect, generated the unary number 01111. We can see from this example how the SP-Multiple-Alignment has captured the recursive nature of the unary number definition.

Figure 53 (b) shows the best SP-Multiple-Alignment produced by the SPCM when '`0`' in New is replaced by the 'axiom' or 'input' string '`0 1 1 1 1`'. This SP-Multiple-Alignment is, in effect, a recognition of the fact that '`0 1 1 1 1`' is a unary number. It corresponds to the way a PCS

```
0                     0                        0
                      |
1                 X a 0 #X                     1
                  |     |
2             X b X     #X 1 #X                2
              |              |
3         X b X              #X 1 #X           3
          |                       |
4     X b X                       #X 1 #X      4
      |                                |
5 X b X                                #X 1 #X 5

(a)

0                     0    1    1    1    1    0
                      |    |    |    |    |
1                 X a 0 #X |    |    |    |    1
                  |     |  |    |    |    |
2             X b X     #X 1 #X |    |    |    2
              |              |  |    |    |
3         X b X              #X 1 #X |    |    3
          |                       |  |    |
4     X b X                       #X 1 #X |    4
      |                                |  |
5 X b X                                #X 1 #X 5

(b)
```

Figure 53: (a) One of many good SP-Multiple-Alignments produced by the SPCM with the SP-Pattern '`0`' in New and the Old SP-Patterns from Figure 52 in Old. (b) The best SP-Multiple-Alignment produced by THE SPCM with '`0 1 1 1 1`' in New and the same SP-Patterns as with Figure 53 (a) in Old.

may be run 'backwards' to recognise input patterns but, since there is no left-to-right arrow in the SP scheme, the notion of 'backwards' processing does not apply. Other unary numbers may be recognised in a similar way.

Section 13.1.4 has described, with examples, how the operation of a PCS in normal form may be understood in terms of the SP theory. Since it is known that any PCS may be modelled by a PCS in normal form ([80], [82, Chapter 13]), we may conclude that the operation of any PCS may be interpreted in terms of the SPTI. Since we also know that any universal Turing machine may be modelled by a PCS [82, Chapter 14] we may also conclude that the operation of any universal Turing machine may be interpreted in terms of the SPTI.

13.2 Potential benefits of the SPTI as a model of computing

Assuming the SPTI can be developed as outlined in Section 13.1, whilst retaining its strengths in intelligence-related aspects of computing and beyond (Section 8), some of the potential benefits are outlined in the following subsections.

13.2.1 Gains in computational efficiency

In Section 2, we saw how, in brains and nervous systems, IC would be favoured by natural selection for the following reasons:

- Direct benefits from IC (Section 2.2):
  - Either by reducing the need for storage of information or by allowing more information to be stored in a given space.
  - Either by reducing the need for large transmission bandwidth or by allowing more information to be transmitted along a given channel.
- And indirect benefits via the close connection between IC and concepts of probability (Section 2.3).

It seems that similar principles may be applied to artificial systems if 'human selection' replaces 'natural selection'. And it seems likely that the benefits outlined above would have a positive impact on 'computational efficiency', broadly defined:

- Gains in terms of the storage or transmission of information may be seen as aspects of computational efficiency.
- Gains via probabilities may be harnessed to speed up calculations, by, for example, choosing the most probable of a set of alternatives before the less probable, where the latter are not needed if any of the most probable options turn out to be right, or at least good enough.

13.2.2 A single, relatively simple language for all computer programs

It is envisaged that, when the SPTI is more mature, the clutter of programming languages in computing will all be replaced by a single, relatively simple language, probably the language of the New Mathematics (Section 11). As described below, this is not as absurd as it may seem.

It is envisaged that, when the SPTI is more mature, that it will be feasible to express all kinds of computer program in a single language composed largely of the eight techniques for IC outlined in Section 4.2, and perhaps other technique later.

This proposal makes sense because there is much similarity amongst existing programming languages, and this is because most of them incorporate the eight techniques for IC:

- Chunking-with-codes can be seen in the usage of named functions or procedures.
- Run-length coding can be seen in iterations like *repeat ... until* and also in recursive programming.
- As mentioned earlier, computational 'objects' are now a feature of most programming languages.
- And more.

13.2.3 A single, relatively simple language for the representation of all kinds of knowledge

As with programming languages, it is envisaged that, when the SPTI is more mature, the clutter of languages for representing knowledge (with a corresponding clutter of 'types' of data) will all be replaced by a single, relatively simple language, probably the language of the New Mathematics (Section 11). For much the same reasons as were described in Section 13.2.2, this is not as absurd as it may seem.

14 Conclusion

The aim of this book is to add substance to the main ideas summarised in the Introduction. In brief:

- The SPTI has been developed with IC at its core because of substantial evidence for the importance of IC in human learning, peception, and cognition (Section 2).

  Since the SPTI draws on evidence for the importance of IC in human learning, peception, and cognition, and since it has things to say about issues in artificial intelligence (AI), it is a theory of both natural and artificial intelligence.
- A working hypothesis in this research is that IC may be achieved via the matching and unification of patterns (ICMUP, Section 4), and more specifically via the concept of SP-Multiple-Alignment (Section 6.3).
- A central part of the SPTI is the SP-Multiple-Alignment concept, a versatile means of compressing information and key to the strengths of the SPTI in diverse aspects of intelligence, in the representation and processing of diverse kinds of intelligence-related knowledge, and in the seamless integration of varied aspects of intelligence, and intelligence-related knowledge, in any combination (Section 8).

  In view of its importance, it is appropriate to repeat that *the SP-Multiple-Alignment concept is a major discovery with the potential to be as significant for an understanding of intelligence as is the concept of DNA for an understanding of biology. It may prove to be the 'double helix' of intelligence!* (Section 6.3.1).

- Since mathematics is the product of human minds and is designed as an aid to human thinking, and in view of the importance of IC in human learning, peception, and cognition, it should not be surprising to find that IC is important in mathematics. It has been demonstrated elsewhere that mathematics may be seen as a set of techniques for the compression of information, and their application, a view of the foundations of mathematics which is radically different from other 'isms' in that area of philosophy.

  Similar arguments can be made about the foundations of logic and computing.

- The idea that IC is central both in the SPTI and in mathematics suggests an amalgamation of the two. That would give the SPTI the benefits of more than 2,000 years of thinking about mathematics. And potential benefits for mathematics and science include: the introduction of new techniques for the representation and processing of knowledge via IC, especially the SP-Multiple-Alignment concept; full or partial automation of inferential processes, and the discovery of new concepts; the potential to provide researchers in mathematics and science with methods for the representation and processing of knowledge that, compared with existing systems, are more compatible with the way that people naturally think.

- There is potential in the SPTI for new thinking about concepts of probability, new thinking about concepts of computation, with potential benefits in both cases.

Comments and questions are very welcome.

This body of interrelated ideas is far from complete. Further research is needed, much of it described in [2] and flagged at appropriate points in the book. I will be happy to discuss possibilities with anyone wishing to investigate these or other issues related to the SP research.

Abbreviations

Abbreviations used in this book are detailed here.

AI	Artificial Intelligence
AIT	Algorithmic Information Theory
AGI	Artificial General Intelligence
DNN	Deep Neural Network
human learning, peception, and cognition	Human Learning, Perception, and Cognition
IC	Information Compression
ICMUP	Information Compression via the Matching and Unification of Patterns
NL	Natural Language
PCS	Post Canonical System
QM	Quantum Mechanics
SPCM	SP Computer Model
SP-Multiple-Alignment	SP-Multiple-Alignment
SPTI	SP Theory of Intelligence

Acknowledgements

A Some key terms

Some key terms, as they are used in this research, are defined here.

Redundancy The term 'redundancy' in a body of information, **I**, means repetition of information in **I**. There is more detail in Appendix B and Section 4.

Information Compression In this book information compression (IC) means 'lossless' compression of information, which means compression of information via reductions in redundancy, *without reductions in non-redundant information.*

The focus on lossless IC is mainly to maintain conceptual simplicity in the SPTI. But it is recognised that, at some stage, there may be a need to consider how lossy IC might be part of the SPTI—perhaps because of evidence for lossy IC in people or other animals, or perhaps to improve the functionality of the SPTI.

Intelligence The term 'intelligence' is used in this book as a shorthand for human intelligence, while 'artificial intelligence' (AI) is the same except that it is artificial and much less fully developed than is intelligence in people.

In both cases, the full meaning covers the kinds of capabilities outlined in Section 8.1 and Section 8.2, and probably more that have not yet been considered.

The Name 'SP' The name 'SP' derives from the concepts of 'Simplicity' and 'Power' which are, as explained in Appendix B, equivalent to the concept of IC, a central part of the SPTI.

Although 'SP derives from 'Simplicity' and 'Power', *it is intended that 'SP' should be treated as a name, in the same way that such names as 'IBM' and 'BBC' are not normally expanded into words.*

B Simplicity and Power

In accordance with Ockham's razor, a good theory should be simple but not so simple that it says little or nothing that is useful.

This may be seen to equate with lossless IC, a process that increases the *Simplicity* ('S') of a body of information, **I**—by the reduction of *redundancy* in **I**—whilst at the same time conserving as much as possible of the non-redundant descriptive or explanatory *Power* ('P') of **I**.

There is relevant discussion in Section 6.7.5.

C The potential risks of artificial intelligence and what can be done about them

It is clear that there are many potential benefits of AI. Nevertheless, from at least as far back as the publication of an article by Irving John Good [83], followed by Nick Bostrom's book on *Superintelligence* [84], there have been worries related to the idea that robots or other manifestations of AI might become more intelligent than people, and if or when that happens, they might then decide that people were no longer needed.

This is a complex subject with no easy answers. No doubt there will be much debate for many years about what may or may not be done to combat risks of that kind. This short appendix makes a few points that may be helpful.

C.1 The possibility of an intelligence explosion

With regard to possible limits to the intelligence of any superintelligence, Bostrom [84, p. 5] says:

> ... however many stops there are between here and human-level machine intelligence, the latter is not the final destination. The next stop, just a short distance farther along the tracks, is superhuman-level machine intelligence. The train might not pause or even decelerate at Humanville Station. It is likely to swoosh right by.

and he quotes with approval what I. J. Good has to say [83, p. 33]:

> Let an ultraintelligent machine be defined as a machine that can far surpass all the intellectual activities of any man however clever. Since the design of machines is one of these intellectual activities, an ultraintelligent machine could design even better machines; there would then unquestionably be an intelligence explosion, and the intelligence of man would be left far behind. Thus the first ultraintelligent machine is the last invention that man need ever make, provided that the machine is docile enough to tell us how to keep it under control.

In short, superintelligence might increase recursively, in what Bostrom calls an 'intelligence explosion', far into the future.

C.2 Even without an intelligence explosion, AIs may be dangerous

In assessing the potential risks of artificial intelligence and what can be done about them, we need to bear in mind that intelligence that is only a little above average human intelligence, coupled with antisocial motivations, can cause havoc, witness the death and destruction caused by Hitler, Pol Pot, Stalin, and others.

It is tempting at first sight to introduce a new term for intelligence that is only a little above average human intelligence. But to keep things simple, that kind of intelligence will be called 'superintelligence', alongside intelligence that is far above human intelligence.

C.3 Possible sources of superintelligence

This section considers three possible sources of superintelligence. A later section discusses how possible risks may be met.

C.3.1 Superintelligence via IC

On the strength of evidence summarised in Sections 6, 8.1 and 8.2, the variety of capabilities that we call intelligence may be understood as IC.

If that is accepted, then with regard to a given body of raw information, **I**, the intelligence of a person or robot may be assessed by the level of compression that may be achieved. But with lossless IC, that measure is limited by the amount of redundancy in **I**, as described in Section B.

When most of the redundancy has been extracted, the person or robot will be reduced to scraping the barrel, and there will be little to choose between different people or robots.

The main point here is that, for a given **I**, there are clear limits to how much lossless compression may be achieved, so that if intelligence is seen to be largely about the levels of IC that may be achieved, there are limits to how intelligent any AI may become. In short, with this view of intelligence, there appears to be no basis for the idea that the creation of a superintelligent AI might lead to a recursive intelligence explosion, as outlined in Appendix C.1.

C.3.2 Superintelligence via speed

A point that, for the sake of clarity, has been omitted from Appendix C.3.1, is that AIs may vary in the speed with which a given level of compression may be achieved with a given body of information **I**.

Since there is in principle no limit to the speed with which any AI may compress information (and thus no limit to the amount of information that may be compressed in a given time), there is clear potential for the kind of intelligence explosion described in Appendix C.1.

C.3.3 Superintelligence via volume

Another factor that may influence the perceived intelligence of an AI is the amount of knowledge which the AI has, and its relevance to the problem or problems at hand—an AI feature that we may call 'volume'.

Unlike people, who can only acquire this kind of knowledge via study or experience, any AI may be supplied directly with volume that, via compression of raw data, has been created by one or more other AIs. Provided it has enough memory, a floor-sweeping robot may be an instant expert on pre-raphaelite art, or civil engineering, and so on.

Since there is in principle no limit to the volume of knowledge of any AI, a concept of intelligence that has volume at centre-stage has clear potential for the kind of intelligence explosion described in Appendix C.1.

C.4 What can be done to avoid the potential risks of superintelligence?

What follows is a few ideas about how the risks of superintelligence may be reduced or eliminated.

C.4.1 Standardisation or limits?

Although it was suggested in Appendices C.3.2 and C.3.3 that, in the regulation of AIs, some kind of standardisation may be required, it may be more appropriate to focus on extremes or limits.

For example, if we are worried about the potential speed with which an AI may compress a given body of data, our main interest is likely to be in the maximum speed that may achieved, not some kind of standard. Likewise, if volume is the worry, we are likely to focus on limits to the amount of knowledge that may be stored, and controls on the kinds of knowledge that may be stored.

C.4.2 Stop research in AI?

If we deliberately fail to reach human-level AI, risks from superintelligent AI could be minimised. Given the strength of the push to develop human-level AI—driven largely by curiosity and the expected benefits of the development—it would be difficult to halt this research.

C.4.3 Take advantage of zero motivations of computers?

Most inanimate entities, including computers, do not have motivations. Hence, they do not have any desire to rule the world, make lots of money, show off in front of their girlfriends, or any of the other of the things that drive people. Hence, if we can develop AIs without ourselves choosing to add motivations, we could be relatively safe from superintellient AI.

A potential problem here is that it is inevitable that any AI with human intelligence or above will develop concepts of motivation from observing how people behave and what they say, by reading news reports, by reading novels and other literary works, and so on. But having a concept of motivation does not in itself mean adopting that motivation—an expert on football does not necessarily want to get on the pitch with world-class players. Issues like these will need careful thought.

C.4.4 An international treaty to control motivations in superintelligences?

The obvious weakness in the proposal in Appendix C.4.3 is that anyone, anywhere, may add motivation to a superintelligent AI. Hence, the problem of superintelligent AIs is not in the superintelligence itself, it is largely the problem of preventing the addition of any motivation to any superintelligence, or tightly controlling what kinds of motivation may be added.

In this connection, Kenan Malik writes:[27]

> ... we already live in societies in which power is exercised by a few to the detriment of the majority, and [AI] provides a means of consolidating that power. ... *There are few tools useful to humans that cannot also cause harm. But they rarely cause harm by themselves; they do so, rather, through the ways in which they are exploited by humans, especially those with power. That, and not fantasy fears of extinction, should be the starting point for any discussion about AI.*

Given the success of the Montreal Protocol to protect the ozone layer, and the success, so far, of agreements designed to minimise the risks of worldwide nuclear war, a treaty to prohibit or limit what motivations may be given to any superintelligence, might allow us to gain the benefits of superintelligence and to minimise the risks. Areas to be considered include:

- Whether or not one may, safely, create superintelligences with relatively benign motives like 'cut the lawn', 'clean the house', and so on?.
- What dangers there may be in motives that appear benign but could lead to problems—the subject of many of Isaac Asimov's robot stories.
- The possible dangers from superintelligences which do not have any motivations in themselves but may provide guidance for 'bad guys' with sinister motivations.

[27] In 'AI doesn't cause harm by itself. We should worry about the people who control it', *The Observer*, 2023-11-26.

C.4.5 The need for transparency in the organisation, workings, and output of any superintelligence

If is accepted that the best way to minimise the risks of any superintelligence would be via some kind of agreement or treaty to constrain what kinds of motivation or motivations, if any, may be added to any superintelligence, an issue that would require clarification would be how to assess any given superintelligence to determine whether or not it conforms to the terms of the treaty.

In that connection, it seems necessary for there to be transparency in the organisation and workings of the superintelligence, and transparency in its output. Without transparency in those aspects, there is potential for unwelcome motivations to be hidden from view within the superintelligence, or within it's operations or outputs.

In that connection, there is a sharp distinction to be made between two kinds of technology which, in the future, are possible bases for the creation of superintelligence:

- *Deep neural networks*. Any superintelligence derived from DNNs may inherit weaknesses in that technology: lack of transparency in how DNNs work, and lack of transparency in their output (Section 8.3.1, item 9).

- *The SPTI*. Any superintelligence derived from the SPTI would probably inherit transparency in its organisation and workings, and transparency in the audit trail that it provides for all its output [65].

D Solomonoff's development of Algorithmic Probability Theory

Solomonoff's research developing APT—about the intimate relationship amongst concepts of IC, inference, and probability—is outlined here.

The theory was first described by Solomonoff in [85]. In a later paper [13], he describes useful background to the research and a relatively informal but arguably more comprehensible description of the theory.

D.1 'Ad-Hoc' and 'Promiscuous' Grammars

In [13, p. 77], Solomonoff writes:

> My main interest ... was learning. I was trying to find an algorithm for the discovery of the 'best' grammar for a given set of acceptable sentences. One of the things I sought was: Given a set of positive cases of acceptable sentences and several grammars, any of which is able to generate all of the sentences, what goodness of fit criterion should be used?

Then he provides a preliminary answer (*ibid.*):

> It is clear that the 'ad-hoc grammar,' that lists all of the sentences in the corpus, fits perfectly. The 'promiscuous grammar' that accepts any conceivable sentence, also fits perfectly. The first grammar has a long description; the second has a short description. It seemed that some grammar half-way between these, was 'correct' but what criterion should be used?

After some detailed discussion, Solomonoff provides an informal summary of a key idea [13, p. 79]:

> At first, most of my evidence for the validity of algorithmic probability was very informal:... It corresponded to (and defined more exactly) the idea of Occam's razor—that 'simple' hypotheses are more likely to be correct.

and later again he quotes Andrey Kolmogorov [13, p. 83]:

> He defined the algorithmic complexity of a string to be the length of the shortest code needed to describe it.

Here, 'algorithmic complexity' may be read as 'information content' and 'code' may be read as 'computer program that describes the string', where the computer is conceived, in APT and AIT, as a universal Turing machine. The 'shortest code' is shorthand for 'the shortest computer program that describes the string that we have been able to find with the time and computational resources that we have available.'

D.2 From information compression to probability

A simplified version of Solomoff's proposals for deriving a measure of probability from a shortest string is also the SPCM method of calculating the absolute probability of each SP-Multiple-Alignment. The calculation of the absolute probability of any SP-Multiple-Alignment is described, with the method of calculating relative probabilities, in Section 6.6.

E Finding good matches between two sequences of symbols

This appendix, based on part of [86],[28] describes the process for finding full matches and good partial matches between two sequences that lies at the heart of the SPCM, and is referenced in Section 4.1. This process was first implemented in SP21, a precursor of SPCM and SP71 that was designed for best-match information retrieval. Much of the discussion is couched in those terms.

E.1 The hit structure

Figure 54 illustrates the main concepts introduced in the description that follows. In this description, the query and the database are both sequences of atomic symbols, assumed to be characters in the discussion, and the database may be divided into sections such as sentences or paragraphs.

Here is the process:

1. The query is processed left to right, one character at a time.
2. Each character in the query is, in effect, broadcast to every character in the database to make a yes/no match in each case.
3. Every positive match (hit) between a character from the query and a character in the database is recorded in a data structure which will be referred to as the *hit structure*:

```
A query:    A B C
            1 2 3

A database:   P A C Q B A B C R
              1 2 3 4 5 6 7 8 9
```

Hit sequences found by SP21 between the query and the database:

```
          A B C
          | | |
P A C Q B A B C R
```

$p_n = 4.630 \text{ x } 10^{-3}$

```
  A         B C
  |         | |
P A C Q B A B C R
```

$p_n = 1.661 \text{ x } 10^{-2}$

```
  A     B     C
  |     |     |
P A C Q B A B C R
```

$p_n = 2.958 \text{ x } 10^{-2}$

```
  A B C
  |   |
P A   C Q B A B C R
```

$p_n = 5.093 \text{ x } 10^{-2}$

The hit structure:

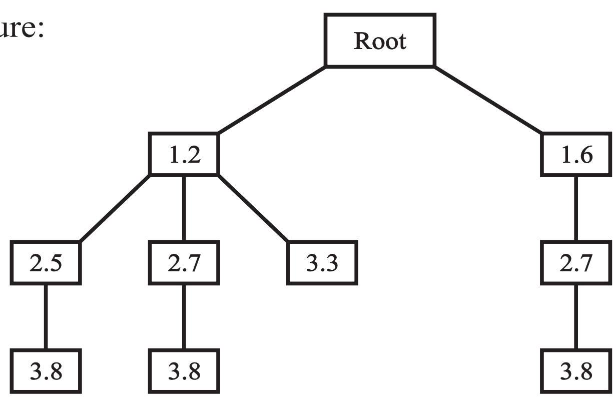

Figure 54: Concepts in pattern matching and search. A 'query' string and a 'database' string are shown at the top with the ordinal positions of characters marked. Sequences of hits between the query and the database are shown in the middle with corresponding values of p_n (described in the text). Each node in the hit structure shows the ordinal position of a query character and the ordinal position of a matching database character. Each path from the root node to a leaf node represents a hit sequence.

(a) The hit structure stores sequences of hits. In each such *hit sequence*, the order of the matched query characters is the same as the order of the matched database characters. But there may be unmatched query characters anywhere within the sequence of matched query characters and there may be unmatched database characters anywhere within the sequence of matched database characters.

(b) The hit structure is implemented as a tree:

- Each path from the root of the tree to any other node records a left-to-right hit sequence, one hit on each node.
- The root of the tree is a dummy node which does not record any hit.

(c) Although a given query character may match two or more characters in the database, only one of these hits is recorded in any one hit sequence. Likewise for database characters.

(d) For each hit recorded in a node in the tree, there is a record of the position of the character in the query, the position of the matching character in the database, and a measure of the probability of the sequence of hits up to and including the given hit (as described below).

(e) If the database is divided into sections, then it can be convenient to apply a rule that all the hits recorded in one path must all come from one section. If this rule is in force (and this may be a decision made by the user) then the system will not attempt to find hit sequences which cross from one section to another.

4. The hit structure is updated every time a hit is found. One or more new nodes for this hit (which will be referred to as the *current hit*) is added to the hit structure in the following way:

   - The tree is examined to identify each hit which immediately precedes the current hit. In this context, the meaning of 'precede' is that the database character for the given hit precedes the database character for the current hit and likewise for the query characters. The meaning of 'immediately' is that, if the node for the given hit has children, then the hits in the child nodes do not precede the current hit. There may, of course, be unmatched query characters or unmatched database characters between the two hits.
   - For each of the 'immediately preceding' hits identified in this way, a probability is calculated for the sequence of hits comprising the path up to and including the current hit.
   - For each hit sequence or path which has been identified in this way, a new node for the current hit is added to the hit structure as the leaf node for that path. In each case, the probability value for the path is recorded in the new node.
   - If there are no paths identified in this way then a new path is started with a node for the current hit as a child of the root and an initial probability value as described below.

5. If the memory space allocated to the hit structure is exhausted at any time then the hit structure is 'purged': the leaf nodes of the tree are sorted in reverse order of their probability values and each leaf node in the bottom half of the set is extracted from the hit structure, together with all nodes on its path which are not shared with any other path. After the hit structure has been purged, the recording of hits may continue using the space which has been released.

6. After the last query character has been processed, the paths from the root to the leaf nodes are displayed in order of their probability in a convenient form for inspection by the user.

E.2 Probabilities

The search process, just described, 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:

- 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 character types used in the query and the database.

- p_1 is the probability of a match between any one character in the query and any one character in the database 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:

$$p\_n = \prod \_i = 1^{i=n}(1 - (1 - p\_1)^{g\_i+1}), \quad g\_1 = 0$$

.

It should be clear from this formula that it is 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.

The thinking behind the method of calculation is straightforward. In accordance with established practice in statistics, the method aims to calculate the probability that the observed distribution of hits, or better, could have occurred by chance on the 'null hypothesis' that all the hits between the query and the database are equi-probable, i.e. that the distribution of hits is random. In this context, a distribution of hits which is 'better' than an observed distribution is one which has more hits within the same range, or the hits fall into clumps, or both these things.

A hit sequence with a low probability is more 'significant' than one with a high probability and may be taken as evidence that the null hypothesis should be rejected. A hit sequence with a low probability is normally more interesting to the user than a high probability hit sequence which could be merely the result of chance.

It is important to stress that this approach to the analysis of probabilities does not in any way prejudge the statistical properties of the query or the database. This is because the focus is on patterns of redundancy *between* the query and the database, not *within* the query or *within* the database. The null hypothesis provides a reference point or baseline for measuring how far an observed distribution of hits between the query and the database departs from randomness. The probability measure that has been described, p_n, is an inverse measure of redundancy *between* the two strings that are being compared. It says nothing about redundancy that may exist within one string or the other, even if one or both of them has as much redundancy as exists in natural languages such as English.

Under the null hypothesis, the probability of an observed hit sequence or better depends on three main factors:

- There is a better chance of finding a hit sequence which is at least as good as the observed sequence if the query or the database or both of them are large.

- If the n hits of a hit sequence are scattered across a relatively long part of the query, or a relatively long part of the database, or both, then the associated probability is higher than for a 'closely packed' hit sequence which is confined to portions of the query and the database which are as short as n or only a little longer.

- Other things being equal, the probability of an observed hit sequence or better decreases as n increases.

For the purposes of information retrieval, the size of the query and the size of the database should not be factors in deciding whether a given hit sequence is significant. What is of interest is the probability of an observed hit sequence after the effects of query size and database size have been abstracted. For this purpose, hit sequences which are closely packed and relatively long are the most significant, independent of the sizes of the query and the database, and independent of where the hit sequences occur within the query and the database.

On this basis, the probability of the first or only hit in a sequence ($p\_1 = 1/A$) is the same as the probability that any given side of an A-sided unbiased die will appear on any one throw of the die. This formula for the first or only hit in a sequence can be derived from the main formula if n is 1 and g_1 is 0. In the main formula, $(1 - p\_1)$ is the probability of a non-match between any one character in the query and any one character in the database. If there are g non-matches between one hit and the next, then the probability of finding one or more hits over a distance of $(g + 1)$ is $(1 - (1 - p\_1)^{g+1})$. This probability is then multiplied by the probability for the hit sequence up to and including the preceding hit, to give p_n.

As an illustration, consider the fourth hit sequence shown in Figure 54. The size of the alphabet, A, is 6, so p_1 is 1/6 which is 0.16, and $(1 - p\_1)$ is 0.83. As with any other hit sequence, the first hit in the sequence has $g\_1 = 0$ so that its probability is $(1-(1-p\_1)^{0+1})$ which is the same as p_1. For the second hit, g_q is 1 and g_d is 0 so that g is 1. The corresponding value of $(1-(1-p\_1)^{1+1})$ is 0.31. This is multiplied by the probability of the hit sequence up to and including the previous hit giving an overall value for p_n of 0.05.

The analysis which has been presented assumes an alphabet of fixed size. This is plausible if the atomic symbols for yes/no matches are characters but may seem less plausible with larger units such as words or phrases because of their variety in natural languages. However, in any one combination of query and database there is a finite (if large) number of different words or phrases. This means that the analysis can be applied even with these larger units.

E.3 Discussion of the search technique

The technique which has been described incorporates the principles of metrics-guided search like this:

- The hit structure plots a set of alternative paths through the search space.

- The probability metric is used (during purging) to prune leaves and branches from the tree of paths.
- The method may be classified as beam search because the search proceeds along several paths at once. This reduces the risk of getting stuck on a local peak. Increasing the amount of memory for the hit structure increases the number of paths and thus increases the chance of finding 'good' hit sequences.

If the database is divided into sections, and if a rule is applied that hits in a hit sequence must all come from the same section, this has the effect of blocking some paths through the search space, thus reducing the number of possibilities which need to be considered and saving some processing time.

There is a trade-off between the maximum size of the hit structure and the ability of the system to find partial matches. When the maximum size of the hit structure is small, processing times are short but the system may get stuck on local peaks and miss partial matches that people can see. When the maximum size of the hit structure is large, the system finds partial matches more effectively but processing times are longer. It seems reasonable that, in a fully-developed version of the system, this trade-off between search time and level of performance should be under the control of the user.

The idea of broadcasting symbols is not in itself especially new and has been described elsewhere [87, 88]. The novelty of the technique which has been described is in the way the broadcasting of symbols is combined with a technique for keeping track of partial matches between the query and the database and in how the system calculates probabilities and uses this information to select amongst the many possible paths through the search space.

In a serial processing environment, the broadcasting of symbols must be done serially, but this kind of operation lends itself very well to the application of parallel processing.

The advantages of this technique compared with the basic dynamic programming method are:

- The space complexity of the process is $O(D)$, better than $O(Q \cdot D)$ for the basic dynamic programming method.
- The method appears to be better suited to parallel processing although, for the approximate string matching problem, an adaptation of dynamic programming for parallel processing has been described [89].
- The technique for pruning the search tree may be applied, however large the search space may be. In general, the 'depth' or thoroughness of searching can be controlled by specifying the maximum size of the hit structure.
- Unlike the standard dynamic programming method, this method can deliver two or more alternative SP-Multiple-Alignments of two patterns.

E.4 Computational complexity

Given the 'combinatorial explosion' of possible matches between two strings, a key question about any system of this kind is the demand which it makes on processing time and computer memory when the quantities of data are increased. This section describes analytic and empirical evidence on these points.

E.4.1 The best, worst and typical cases

The core of the search process is the broadcasting of characters from one string (the query) to each of the characters in another string (the database). From the perspective of absolute running times and computational complexity, the best case is when none of the symbols in the query match any of the symbols in the database. In this case, there are no hit sequences to be stored and the search is completed very quickly.

The worst case is when all the symbols in the query and the database are the same. In principle, this yields the largest possible number of hit sequences although, in practice, SPCM will purge many of them from its hit structure.

The typical case, somewhere between the two extremes, is where the query and the database both contain a range of alphabetic symbol types distributed in the kind of way that letters are distributed in natural languages.

Since the worst case is unlikely to occur in practice, the typical case has been assumed in what follows.

E.4.2 Time complexity in a serial processing environment

In a serial processing environment, it is clear that the processing time for this operation is proportional to the length of the query string and, independently, it is proportional to the length of the database. In other words, the time complexity for this operation is $\mathrm{O}(n \cdot m)$, where n is the number of characters in the query and m is the number of characters in the database.

E.4.3 Updating the hit structure

The process of updating the hit structure includes the time required to search the hit structure for the best hit sequences and the time required to add new nodes. The time required to search the hit structure will vary, depending on whether the hit structure is full or has recently been purged; but, apart from a small effect at the start of processing as the space available for the hit structure is filled, the time required for this operation should be independent of n or m.

Since the updating operation occurs only for hits, and since the proportion of hits amongst the yes/no matches should be independent of n or m, we may conclude, overall, that our initial assessment of the algorithm remains valid. In short, an analysis of the algorithm shows that its time complexity in a serial processing environment should be $\mathrm{O}(n \cdot m)$. This analysis is independent of the size of the hit structure.

The foregoing analysis remains valid when the database is divided into sections with the exclusion of hit sequences from one section to another. This kind of constraint can save overall processing time by reducing the variety of hit sequences and thus reducing the number of purges of the hit structure; but the constraint does not change the relationship between processing time and n or m.

E.4.4 Time complexity in a parallel processing environment

As previously noted, the search process lends itself well to parallel processing:

- The process of broadcasting a query character to every character in the database is an intrinsically parallel operation.

- If the database is divided into parts, each part with its own small hit structure, then updating of the hit structures may be performed in parallel.

If finding hits and updating the hit structure takes unit time independent of the size of the database, as seems possible in a parallel processing environment, then the time complexity of the process should be $O(n)$.

E.4.5 Space complexity

The space required to store the database is independent of any retrieval mechanism and is therefore excluded from this analysis of the space complexity of the search process. At this level of abstraction, there is no distinction between 'main memory' and 'secondary storage', since both kinds of memory are assumed to function as a unified 'virtual memory'. The memory required specifically for the search process is mainly the space required to store the hit structure.

Although the hit structure varies in size as the program runs, it never exceeds a pre-defined limit because it is purged whenever the limit is reached. If the user requires all hit sequences down to a fixed level of 'quality' then, for typical data, the size of the hit structure should be increased in proportion to m and the space complexity of the process would be $O(m)$.

E.4.6 Empirical evidence

Running times for SP21 have been plotted to show the effect of varying the size of the query (with a database of constant size) and also to show the effect of varying the size of the database (with a query of constant size) [86]. The queries and the databases were all samples of English.

In each case, the relationship is approximately linear. These results lend support to the analytic conclusion that the time complexity of the SP21 process in a serial processing environment is $O(n \cdot m)$.

F Redundancy is often useful in the detection and correction of errors and in the storage and processing of information

The fact that redundancy—repetition of information—is often useful in the detection and correction of errors and in the storage and processing of information, and the fact that these things are true in biological systems as well as artificial systems, is the second apparent contradiction to the SPTI as a theory of human learning, peception, and cognition. Here are some examples:

- *Backup copies.* With any kind of database, it is normal practice to maintain one or more backup copies as a safeguard against catastrophic loss of the data. Each backup copy represents redundancy in the system.

- *Mirror copies.* With information on the internet, it is common practice to maintain two or more mirror copies in different places to minimise transmission times and to spread processing loads across two or more sites, thus reducing the chance of overload at any one site. Again, each mirror copy represents redundancy in the system.

- *Redundancies as an aid to the correction of errors.* Redundancies in natural language can be a very useful aid to the comprehension of speech in noisy conditions.

- *Redundancies in electronic messages.* It is normal practice to add redundancies to electronic messages, in the form of additional bits of information together with checksums, and also by repeating the transmission of any part of a message that has become corrupted. These things help to safeguard messages against accidental errors caused by such things as birds flying across transmission beams, or electronic noise in the system, and so on.

In information processing systems of any kind, uses of redundancy of the kind just described may co-exist with ICMUP. For example: '... it is entirely possible for a database to be designed to minimise internal redundancies and, at the same time, for redundancies to be used in backup copies or mirror copies of the database ... Paradoxical as it may sound, knowledge can be compressed and redundant at the same time.' [1, Section 2.3.7].

G Heuristic search

With most intelligence-related programs, there is a target problem to be solved but the number of possible solutions is far too large for a solution to the target problem to be found via an exhaustive search of the possible solutions.

In cases like these, it is necessary to adopt an heuristic search strategy. This means searching the space of possible solutions and partial solutions in stages, and, at the end of each stage, choosing the most promising partial solutions for further development. Those kinds of technique include hill climbing, genetic algorithms, simulated annealing, and more.

With this kind of strategy, it is normally possible to find acceptably good solutions within a reasonable time, and with an acceptable computational complexity, but it is not normally possible to guarantee that the best possible solution has been found. allows a New SP-Pattern (sometimes more than one) in row 0 (or column 0) to be encoded economically in terms

In the SPTI, this kind of strategy is normally required in two areas:

- In the building of one or more SP-Multiple-Alignments, each one of whichof the Old SP-Patterns in the remaining rows (or columns), one SP-Pattern per row (or column).

- In the creation of one or more SP-grammars, each one of which is a set of Old SP-Patterns that provides for the economical encoding of a relatively large body of New information.

H A note on copyright

Because this book is about the SPTI and the SPCM, most of it is drawn from existing peer-reviewed publications. These are mostly in open access journals with CC-BY licences, except from my previous book [1], for which I was both the author and the publisher. So in general, while the reuse of figures and text from my own research publications is acknowledged, it has not been necessary to seek permission for its reuse.

References

[1] J. G. Wolff, *Unifying Computing and Cognition: the SP Theory and Its Applications.* Menai Bridge: CognitionResearch.org, 2006, distributors include Amazon.com and Amazon.co.uk.

The print version is produced via print-on-demand from "INGRAM Lightning Source" and, via that technology, is unlikely to go out of print.

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

[3] J. G. Wolff, "The SP Theory of Intelligence: an overview," *Information*, vol. 4, no. 3, pp. 283–341, 2013, arXiv:1306.3888 [cs.AI]. bit.ly/1NOMJ6l.

[4] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] R. Shwartz-Ziv and Y. LeCun, "To compress or not to compress—self-supervised learning and information theory: a review," vol. arXiv:2304.09355v3 [cs.LG], pp. 1–39, 2023, arxiv.

[6] J. G. Wolff, "Twenty significant problems in AI research, with potential solutions via the SP Theory of Intelligence and its realisation in the SP Computer Model," *Foundations*, vol. 2, pp. 1045–1079, 2022.

[7] ——, "Big data and the SP Theory of Intelligence," *IEEE Access*, vol. 2, pp. 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.

[8] ——, "Mathematics as information compression via the matching and unification of patterns," *Complexity*, vol. 2019, p. 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.

[9] ——, "The SP Theory of Intelligence, and its realisation in the SP Computer Model as a foundation for the development of artificial general intelligence," *Analytics*, vol. 2, no. 1, pp. 163–192, 20023, web: www.mdpi.com/2813-2203/2/1/10.

[10] ——, "Information compression, multiple alignment, and the representation and processing of knowledge in the brain," *Frontiers in Psychology*, vol. 7, no. 1584, 2016, arXiv:1604.05535 [cs.AI], bit.ly/2esmYyt.

[11] ——, "Information compression as a unifying principle in human learning, perception, and cognition," *Complexity*, vol. 2019, p. 38, February 2019, article ID 1879746. viXra:1707.0161v3, hal-01624595 v2.

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

[13] R. J. Solomonoff, "The discovery of algorithmic probability," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 73–88, 1997.

[14] V. Balasubramanian, "Brain power," *Proceedings of the National Academy of Sciences USA*, vol. 118, no. 32, 2021, published online.

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

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

[17] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, pp. 183–193, 1954.

[18] ——, *Applications of Information Theory to Psychology.* New York: Holt, Rinehart and Winston, 1959.

[19] H. B. Barlow, "Trigger features, adaptation and economy of impulses," in *Information Processes in the Nervous System*, K. N. Leibovic, Ed. New York: Springer, 1969, pp. 209–230.

[20] N. Chater, "Reconciling simplicity and likelihood principles in perceptual organisation," *Psychological Review*, vol. 103, no. 3, pp. 566–581, 1996.

[21] N. Chater and P. Vitányi, "Simplicity: a unifying principle in cognitive science?" *TRENDS in Cognitive Sciences*, vol. 7, no. 1, pp. 19–22, 2003.

[22] A. S. Hsu, N. Chater, and P. Vitányi, "Language learning from positive evidence, reconsidered: a simplicity-based approach," *Topics in Cognitive Science*, vol. 5, pp. 35–55, 2013.

[23] H. B. Barlow, "Redundancy reduction revisited," *Network: Computation in Neural Systems*, vol. 12, pp. 241–253, 2001.

[24] J. G. Wolff, "Learning syntax and meanings through optimization and distributional analysis," in *Categories and Processes in Language Acquisition*, Y. Levy, I. M. Schlesinger, and M. D. S. Braine, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1988, pp. 179–215, bit.ly/ZIGjyc.

[25] M. Gold, "Language identification in the limit," *Information and Control*, vol. 10, pp. 447–474, 1967.

[26] J. G. Wolff, "Language acquisition and the discovery of phrase structure," *Language & Speech*, vol. 23, pp. 255–269, 1980.

[27] A. Newell, "You can't play 20 questions with nature and win: Projective comments on the papers in this symposium," in *Visual Information Processing*, W. G. Chase, Ed. New York: Academic Press, 1973, pp. 283–308.

[28] A. Newell, Ed., *Unified Theories of Cognition.* Cambridge, Mass.: Harvard University Press, 1990.

[29] J. E. Laird, *The Soar Cognitive Architecture.* Cambridge, Mass.: The MIT Press, 2012, ISBN-13: 978-0-262-12296-2.

[30] P. McCorduck, *Machines Who Think: a Personal Inquiry Into the History and Prospects of Artificial Intelligence*, 2nd ed. Natick, MA: A. K. Peters Ltd, 2004.

[31] J. E. Kelly and S. Hamm, *Smart Machines: IBM's Watson and the Era of Cognitive Computing*, Kindle ed. New York: Columbia University Press, 2013.

[32] H. L. Dreyfus, *What Computers Can't Do: A Critique of Artificial Reason.* New York: Harper and Row, 1972, revised, 1979.

[33] G. F. Marcus and E. Davis, *Rebooting AI: Building Artificial Intelligence We Can Trust*, Kindle ed. New York: Pantheon Books, 2019.

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

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

[36] J. G. Wolff, "The *sp-multiple-alignment* concept as a generalization of six other variants of information compression via the matching and unification of patterns," *Journal of Pure and Applied Mathematics*, vol. 7, no. 4.

[37] ——, "Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision," *SpringerPlus*, vol. 3, no. 1, pp. 552–570.

[38] N. S. Yanofsky and M. Zelcer, "The role of symmetry in mathematics," 2016, arXiv:1502.07803v2 [math.HO].

[39] J. Lyons, *Introduction to Theoretical Linguistics.* Cambridge: Cambridge University Press, 1968.

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

[41] M. Bramer, *Logic Programming With Prolog*, second, Kindle ed. London: Springer-Verlag, 2013.

[42] C. Brown, *My Left Foot*, Kindle ed. London: Vintage Digital, 2014, first published in 1954.

[43] E. H. Lenneberg, "Understanding language without the ability to speak: a case report," *Journal of Abnormal and Social Psychology*, vol. 65, pp. 419–425, 1962.

[44] N. Chomsky, *Aspects of the Theory of Syntax.* Cambridge, MA: MIT Press, 1965.

[45] J. G. Wolff, "Probabilistic reasoning as information compression by multiple alignment, unification and search," School of Informatics, University of Wales at Bangor, Tech. Rep., 1998, see www.cognitionresearch.org/papers/pr/pr.htm.

[46] S. J. D. Prince, *Computer Vision: Models, Learning, and Inference.* Cambridge: Cambridge University Press, 2012, ISBN 978-1-107-01179-3.

[47] J. G. Wolff, "Towards an intelligent database system founded on the SP Theory of Computing and Cognition," *Data & Knowledge Engineering*, vol. 60, pp. 596–624, 2007, arXiv:cs/0311031 [cs.DB], bit.ly/1CUldR6.

[48] N. Chomsky, *Syntactic Structures.* The Hague: Mouton, 1957.

[49] F. C. N. Pereira and D. H. D. Warren, "Definite Clause Grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks," *Artificial Intelligence*, vol. 13, pp. 231–278, 1980.

[50] G. Gazdar and C. Mellish, *Natural Language Processing in Prolog.* Wokingham: Addison-Wesley, 1989.

[51] J. G. Wolff, "Cognitive development as optimization," in *Computational Models of Learning*, L. Bolc, Ed. Heidelberg: Springer-Verlag, 1987, pp. 161–205.

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

[53] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, revised second printing ed. San Francisco: Morgan Kaufmann, 1997.

[54] T. G. Evans, "A program for the solution of a class of geometric-analogy intelligence-test questions," in *Semantic Information Processing*, M. L. Minsky, Ed. Cambridge Mass.: MIT Press, 1968, pp. 271–353.

[55] T. Belloti and A. Gammerman, "Experiments in solving analogy problems using Minimal Length Encoding," in *Proceedings of Stream 1: "Computational Learning and Probabilistic Reasoning", Applied Decision Technologies, Brunel University, April 1995*, 1996.

[56] A. J. Gammerman, "The representation and manipulation of the algorithmic probability measure for problem solving," *Annals of Mathematics and Artificial Intelligence*, vol. 4, pp. 281–300, 1991.

[57] J. G. Wolff, "The SP Theory of Intelligence: benefits and applications," *Information*, vol. 5, no. 1, pp. 1–27, 2014, arXiv:1307.0845 [cs.AI], bit.ly/1FRYwew.

[58] ——, "Software engineering and the SP Theory of Intelligence," CognitionResearch.org, Tech. Rep., 2017, arXiv:1708.06665 [cs.SE], bit.ly/2w99Wzq.

[59] J. Schmidhuber, "One big net for everything," The Swiss AI Lab, IDSIA, Tech. Rep., 2018.

[60] J. G. Wolff, "Interpreting Winograd Schemas via the SP Theory of Intelligence and its realisation in the SP Computer Model," CognitionResearch.org, Tech. Rep., 2018, bit.ly/2ME8DOA.

[61] ——, "A proposed solution to problems in learning the knowledge needed by self-driving vehicles," CognitionResearch.org, Tech. Rep., 2021, submitted for publication, tinyurl.com/n6vpxcuf.

[62] E. Davis and G. Marcus, "Commonsense reasoning and commonsense knowledge in artificial intelligence," *Communications of the ACM*, vol. 58, no. 9, pp. 92–103, 2015.

[63] J. G. Wolff, "Commonsense reasoning, commonsense knowledge, and the SP Theory of Intelligence," CognitionResearch.org, Tech. Rep., 2019, viXra:1901.0051v2, hal-01970147 version 3, bit.ly/2RESeut.

[64] ——, "How the SP System may promote sustainability in energy consumption in IT systems," *Sustainability*, vol. 13, no. 8, pp. 1–21, 2021. [Online]. Available: https://www.mdpi.com/2071-1050/13/8/4565

[65] ——, "Transparency and granularity in the SP Theory of Intelligence and its realisation in the SP Computer Model," in *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, W. Pedrycz and S.-M. Chen, Eds. Heidelberg: Springer, 2021, iSBN 978-3-030-64948-7, arXiv:2009.06370 [cs.AI].

[66] ——, "The potential of the SP System in machine learning and data analysis for image processing," *Big Data and Cognitive Computing*, vol. 5, no. 1, p. 7, 2021, tinyurl.com/7y9fbe7f.

[67] P. Jonas and G. Buzsaki, "Neural inhibition," *Scholarpedia*, vol. 2, no. 9, p. 3286, 2007.

[68] H. K. Hartline and F. Ratliff, "Inhibitory interaction of receptor units in the eye of *limulus*," *Journal of General Physiology*, vol. 40, pp. 357–376, 1957.

[69] H. K. Hartline, H. G. WAGNER, and F. Ratliff, "Inhibition in the eye of *limulus*," *Journal of General Physiology*, vol. 39, no. 5, pp. 651–673, 1956.

[70] L. R. Squire, D. Berg, F. E. Bloom, S. du Lac, A. Ghosh, and N. C. Spitzer, Eds., *Fundamental Neuroscience*, 4th ed. Amsterdam: Elsevier, 2013.

[71] G. J. Chaitin, "Randomness in arithmetic," *Scientific American*, vol. 259, no. 1, pp. 80–85, 1988.

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

[73] E. Wigner, "The unreasonable effectiveness of mathematics in the natural sciences," *Communications in Pure and Applied Mathematics*, vol. 13, pp. 1–14, 1960, richard Courant Lecture in Mathematical Sciences delivered at New York University, May 11, 1959.

[74] M. Chown, *The Magicians: Great Minds and the Central Miracle of Science*, Kindle ed. London: Faber & Faber Ltd, 2020.

[75] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 49, no. 9, pp. 1098–1101, 1952.

[76] A. M. Turing, "On computable numbers with an application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. 42, pp. 230–265 and 544–546, 1936.

[77] J. B. Rosser, "Highlights of the history of the lamda-calculus," *Annals of the History of Computing (USA)*, vol. 6, no. 4, pp. 337–349, 1984.

[78] S. C. Kleene, "l-definability and recursiveness," *Duke Mathematical Journal*, vol. 2, pp. 340–353, 1936.

[79] A. A. Markov and N. M. Nagorny, *The Theory of Algorithms.* Dordrecht: Kluwer, 1988.

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

[81] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[82] M. L. Minsky, *Computation, Finite and Infinite Machines.* Englewood Cliffs, NJ.: Prentice Hall, 1967.

[83] I. J. Good, "Speculations concerning the first ultraintelligent machine," in *Advances in Computers*, F. L. Alt and M. Rubinoff, Eds. New York: Academic Press, 1965, vol. 6, p. 3188.

[84] N. Bostrom, *Superintelligence*, Kindle ed. Oxford: Oxford University Press.

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

[86] J. G. Wolff, "A scaleable technique for best-match retrieval of sequential information using metrics-guided search," *Journal of Information Science*, vol. 20, no. 1, pp. 16–28, 1994, see www.cognitionresearch.org/papers/ir/ir.htm.

[87] D. M. Carroll, C. A. Pogue, and P. Willett, "Bibliographic pattern matching using the ICL Distributed Array Processor," *Journal of the American Society for Information Science*, vol. 39, no. 6, pp. 390–399, 1988.

[88] D. L. Lee, "ALTEP - a cellular processor for high-speed pattern matching," *New Generation Computing*, vol. 4, no. 3, pp. 225–244, 1986.

[89] A. A. Bertossi, F. Luccio, L. Pagli, and E. Lodi, "A parallel solution to the approximate string matching problem," *Computer Journal*, vol. 35, no. 5, pp. 524–526, 1992.