FOUNDATIONS OF INTELLIGENCE VIA INFORMATION COMPRESSION: THE SP THEORY

J Gerard Wolff*

Sunday 8th June, 2025

Abstract

This book is about the *SP Theory of Intelligenc* (SPTI) and its realisation in the *SP Computer Model* (SPCM). The SPTI draws on substantial evidence for the importance of information compression (IC) in human learning, perception, and cognition, and is thus a theory of both natural and artificial intelligence. In the SPTI, IC is achieved largely via the powerful concept of *SP-Multiple-Alignment* (SPMA), *a major discovery* which is largely responsible for the versatility of the SPTI in diverse aspects of intelligence, and the seamless integration of diverse aspects of intelligence in any combination, a necessary feature of any system that aspires to human-level intelligence. Also, it provides, paradoxically, for both the compression and decompression of information. These ideas have led to *a second major discovery*, that much of mathematics, perhaps all of it, may be seen as a collection of mechanisms for IC, and their application. This observation suggests the creation of a *New Mathematics* (NM) 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 is also relevant to areas less closely associated with AI. These include: the management of 'big data'; medical databases; sustainability of computing; transparency in computing; and computer vision.

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

Contents

1 Introduction **5**
- 1.1 What is artificial human-level intelligence or AGI? . . . 6
- 1.2 My background and experience . . . 6
- 1.3 Alternative perspectives on AI and models of human cognition . . . 7
- 1.4 Presentation . . . 13
- 1.5 A note on copyright . . . 13

2 Unique selling points **14**
- 2.1 Main USPs . . . 14
- 2.2 Other USPs . . . 14

3 In the quest for artificial general intelligence, the importance of research strategy **16**
- 3.1 Problems with a depth-first strategy in the development of theory . 16
- 3.2 The benefits of a top-down research strategy . . . 19
- 3.3 The adoption of a top-down research strategy in the SP research . . 21

4 Information compression in human learning, perception, and cognition **21**
- 4.1 Pioneering research by Fred Attneave, Horace Barlow, and Satosi Watanabe . . . 22
- 4.2 Storage and transmission of information . . . 24
- 4.3 Merging multiple views to make one . . . 25
- 4.4 Binocular vision . . . 25
- 4.5 IC and concepts of prediction and probability . . . 26
- 4.6 Research on the learning of a first language . . . 28
- 4.7 Barlow's change of view . . . 32

5 The origin and development of the SP Theory of Intelligence **33**
- 5.1 Early ideas . . . 33
- 5.2 Getting a grip on the problem . . . 33
- 5.3 The potential of multiple sequence alignments (MSAs) or something like them . . . 34
- 5.4 Developing the SP-Multiple-Alignment concept . . . 35

6 The SP Theory of Intelligence and its realisation in the SP Computer Model **36**
- 6.1 High level view of the SPTI . . . 36
- 6.2 *SP-Patterns* and *SP-Symbols* . . . 37
- 6.3 The SP-Multiple-Alignment concept . . . 38

6.4 Unsupervised learning in people and other animals 58
6.5 The SP Computer Model 70
6.6 *SP-Neural*: a preliminary version of the SP Theory of Intelligence in terms of neurons and their inter-connections and inter-communications 70
6.7 Future developments and the SP Machine 72

7 Examples of the versatility of the SP-Multiple-Alignment concept within the SPCM 74
7.1 Recursive processing and the SP Theory of Intelligence 74
7.2 Ambiguities in language 76
7.3 Robustness in the face of errors of omission, addition, and substitution 77
7.4 Syntactic dependencies in French 79
7.5 Dependencies in the syntax of english auxiliary verbs 83
7.6 The integration of syntax with semantics 92
7.7 Recognition and retrieval 97
7.8 Medical diagnosis 99
7.9 Nonmonotonic reasoning and reasoning with default values 101
7.10 Explaining away 'explaining away': The SP Theory of Intelligence as an alternative to Bayesian networks 105
7.11 Planning 117
7.12 Problem solving 120

8 Aspects of the SPTI 127
8.1 Features of the SPTI and what they are good for 127
8.2 The seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination 128
8.3 The clear potential of the SPTI to solve 23 significant problems in AI research 129
8.4 The SPTI as a foundation for the development of artificial general intelligence 134
8.5 Commonsense reasoning and commonsense knowledge 134
8.6 Papers about potential benefits and applications of the SPTI 134
8.7 How one mechanism may achieve both the production and the analysis of data 136

9 Summary of the strengths of the SPTI and its realisation in the SPCM 136
9.1 Summary of AI-related strengths of the SPTI 136
9.2 Summary of strengths of the SPTI less closely-related to AI 138

10 Information compression provides an entirely novel perspective on the foundations of mathematics, logic, and computing 139
10.1 Chunking-with-Codes in mathematics . . . 139
10.2 Schema-Plus-Correction in mathematics . . . 140
10.3 Run-length-coding in mathematics . . . 141
10.4 Combinations of these techniques within mathematics . . . 141
10.5 Logic and computing . . . 142
10.6 Mathematics, information compression, and probabilities . . . 142
10.7 Why is mathematics so unreasonably effective in the natural sciences? 143

11 A 'New Mathematics' as an integration of mathematics with the SP Theory of Intelligence 144
11.1 The potential of the SPTI as a theory of probabilities . . . 145
11.2 The potential of the SPTI as a theory of computing . . . 148

12 Conclusion 153

A Some key terms 156

B Simplicity and Power 157
B.1 Aim to make **I** as large as possible . . . 158
B.2 Comparing one system with another . . . 158
B.3 'Dirty data' . . . 158
B.4 The main ideas in this book . . . 158

C The potential risks of artificial intelligence and what can be done about them 161
C.1 The possibility of an intelligence explosion . . . 161
C.2 Even without an intelligence explosion, AIs may be dangerous . . . 162
C.3 Possible sources of super-intelligence . . . 162
C.4 For people, the risks of super-inrelligence . . . 163
C.5 What can be done to avoid the potential risks of super-intelligence? 164

D Solomonoff's development of Algorithmic Probability Theory (APT) 166
D.1 'Ad-Hoc' and 'Promiscuous' Grammars . . . 167
D.2 From information compression to probability . . . 168
D.3 Finding good matches between two sequences of SP-Symbols . . . 168
D.4 Probabilities . . . 171
D.5 Discussion of the search technique . . . 174
D.6 Computational complexity . . . 175

E Redundancy is often useful in the detection and correction of errors and in the storage and processing of information 177

F Heuristic search 178

G The working hypothesis that information compression may always be achieved via the full or partial matching and unification (merging) of patterns (ICMUP) 179
G.1 Searching for repeating patterns 180
G.2 Eight techniques for ICMUP 181
G.3 Is ICMUP the same as 'symmetry'? 186

H Concepts of computing and the Post Canonical System (PCS 187
H.1 The structure and workings of a PCS 188
H.2 How the PCS works 188
H.3 With the PCS, the creation and recognition of numbers in unary notation 189

I Big tech companies, AI, and intellectual property 190

J Barlows change of view about the significance of IC in mammalian learning, perception, and cognition, with comments 191

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 book about ideas previously described in [86] together with ideas from later publications and from new thinking. In parts, it draws on [86], but it is free-standing with new perspectives and much that was not in the earlier book.

Since the development of the SPTI draws on evidence relating to natural 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 (Chapter 6.6), and somewhat unexpectedly, the foundations of mathematics (Chapter 10).

Since the SPTI is not a comprehensive theory of what it aims to describe—the nature of intelligence—the SPTI should best be regarded as a *foundation* for the development of human-level artificial intelligence, aka 'artificial general intelligence' (AGI), and that further development will be needed, as outlined in [53], before the SPTI can be described as a full theory of AGI (Section 8.4).

1.1 What is artificial human-level intelligence or AGI?

Before we proceed, a few words are needed about the nature of the ultimate goal of the research: artificial human-level intelligence or AGI:

- In brief, there is much more to natural human-level intelligence than such things as computers that can beat the best human players of Go [20, 69], or computers that can, fast and accurately, fold sequences of amino-acid residues into 3D protein structures [70, 73].

- A major distinguishing feature of natural human-level intelligence is its ability to find new solutions to previously unfamiliar problems, not the relatively narrow capabilities of any one or more of the kinds of applications mentioned above (Section 3.1.6).

  Natural human-level intelligence has a fluidity and versatility that is largely missing from specific applications of AI.

- Another feature of natural human-level AI is the surprisingly challenging problem of modelling commonsense reasoning and commonsense knowledge [22].

1.2 My background and experience

As a possible aid to understanding what follows, here is a brief summary of my background and experience, and a few details about how the SPTI was developed.

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. My PhD is summarised in [80].

After those beginnings, 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, Praxis Systems plc, in the city of Bath.

After gaining valuable experience in software engineering, I returned to academic work in 1988, 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 Computer Science and Electronic Engineering in Bangor University). The research has continued within the non-profit *CognitionResearch.org* following my application for, and acceptance of, early retirement within the University's ER scheme.

By early 2006, I had completed a substantial book about the research ([86]) but I failed to publicise it because concerns about climate change led me into nearly seven years' of full-time campaigning, chiefly assisting Phil Thornhill with organising annual marches for action on climate change, and later promoting Dr Gerhard Knies's 'Desertec' concept in the UK (`https://en.wikipedia.org/wiki/Desertec`). That concept evolved from research by the "German Aerospace Center (DLR) between 2004 and 2007 demonstrated that the desert sun could meet rising power demand in the MENA region (Middle East and North Africa) while also helping to power Europe, reduce carbon emissions across the EU-MENA region and power desalination plants to provide freshwater to the MENA region." (quoted from the Wikipedia article about Desertec).

In late 2012, I returned to work on the SPTI. To help raise awareness of the SPTI ideas, I published a shortened version of [86] in a peer-reviewed article [89]. In years following, I have developed the 'Mathematics as IC' idea (Chapter 10, [100]), and I have published other peer-reviewed journal articles, mainly about the potential benefits and applications of the SPTI. Most of these may be downloaded via links from `https://tinyurl.com/3p7speet`.

1.3 Alternative perspectives on AI and models of human cognition

This section summarises research relating to the role of IC in AI and in models of human cognition, including: a variety of studies in those areas; Algorithmic Information Theory (AIT); Bayes' theorem; and deep neural networks (DNNs).

This section also includes a brief discussion in Section 1.3.4 of why mathematics is not more prominent in the book (although it is used in many parts of the SPCM as may be seen in [99, Appendix A], and it is central in 'Mathematics as IC', Chapter 10, [100]).

The idea that IC might be important in aspects of AI or models of human cognition has been developed in various studies detailed here, with some comments about relationships with the SPTI:

- As outlined in Section 4.1, the approach to AI/cognition described in this book was begun by several researchers, most notably Fred Attneave [1, 2],

Horace Barlow [3, 4], and Satosi Watanabe [74, 75].

As noted in Section 4.1.3, at a remarkably early stage, Barlow made the prescient observation that IC might be closely related to natural intelligence.

- *Solomonoff's Algorithmic Probability Theory.* IC has a central role in Solomonoff's APT (Appendix D).

  Solomonoff's APT is itself part of the foundations of Algorithmic Information Theory (AIT [44]). A central idea here is that the algorithmic information of a body of information, **I**, is the length of the smallest program that anyone has managed to create or discover that can create **I**.

  Here, the definition of 'program' is expressed in terms of Alan Turing's model of computing. This is quite different from the SPTI which is founded on concepts of ICMUP (Appendix G) and SP-Multiple-Alignment (Section 6.3).

- The SPTI grew out of an earlier programme of research on a young child's learning of his or her first language (Section 4.6, [80]). A main conclusion from that research is the central importance of IC in the learning of a first language.

- More generally, I proposed IC as the basis for a general theory of computing and cognition [81].

- Chater and Vityányi [15] propose data compression as a unifying principle in cognitive science. The framework they develop is quite different from the framework of the SPTI.

- Marcus Hutter [37] views data compression as the key to a mathematical definition of intelligence.

- Jürgen Schmidhuber [60] suggests using Kurt Gdel's self-reference trick of 1931 (constructing a sentence that, when evaluated, indirectly refers to itself) as a way of assessing the efficiency of a program's method for solving problems.

- In another paper, [61], Schmidhuber reviews research on learning in DNNs. The review has a few short sections about the possible role for IC in DNN learning, without any recognition that IC may be fundamental in *all* aspects of intelligence, as it is in the SPTI.

  This book includes some discussion in Section 1.3.3 of what Schmidhuber says about possible roles for IC in learning in DNNs.

- Phil Maguire and colleagues [46] describe how a theory of mind may be based on data compression.

- Ravid Shwartz-Ziv and Yann LeCun [65] provide a wide-ranging review of the intersection of information theory, self-supervised learning, and DNNs, aiming for a better understanding through their proposed unified approach.

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 the body of inter-related ideas associated with such names as the aforementioned AIT, 'Algorithmic Probability Theory' (APT), 'Minimum Description Length', 'Minimum Message Length', and 'Kolmogorov Complexity' [44]. In the rest of this book, all these ideas are referred to collectively as 'AIT', except in Appendix D where the focus is on Solomonoff's 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 compression of information 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, and strengths in other areas (Chapter 9).

- 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 'ICMUP', within the SPTI (Appendix G).

- That the importance of IC in the SPTI suggests new foundations for mathematics which are radically different from any of the existing 'isms' in the foundations of mathematics (Chapter 10, [100]).

- There is potential for the SPTI to be integrated with mathematics, thus creating a *New Mathematics* with many potential advantages (Chapter 11);
- The SPTI has potential as the basis for new concepts of probability (Section 11.1);
- And the SPTI has potential as the basis for a new model of computing (Section 11.2).
- By contrast with AIT, in which there is research relating AIT to Bayes' theorem (see, for example, [44, pp. 350–353] and APT [67, 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.
- $P(B|A)$ 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 4.6 and Section 6.4.7.

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

- 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 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, perception, and cognition (Chapter 4), 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 Section 7.10 and [86, 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 weaknesses in DNNs.

With respect to the empirical underpinnings of the SPTI as it is now, and likely to develop in the future, DNNs have several shortcomings. Several of these shortcomings, 21 in all, are marked with '`[DNN]`' in a list of 23 problems and the potential for their solution via the SPTI in Section 8.3.

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.'" [58, 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 F), the SP-Multiple-Alignment concept (Section 6.3), and unsupervised learning (Section 6.4).

- 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 mathematics, especially when mathematics is created and evaluated with unassisted brain power. That said, mathematics has been included here and there in the SPCM.

In general the advantages of expressing theories with programs and not exclusively with mathematics is that programs provide more scope for expressing new ideas.[2]

In general, the advantages of creating and running programs as outlined above include:

- Like mathematics, programs enforce rigour in the expression of concepts but arguably programs can express a much wider variety of concepts than mathematics.

- Unlike people, computers don't get tired and make mistakes.

- Computers can calculate much faster than people.

- And programs facilitate the testing of new ideas, and demonstrating how a theory works.

For those kinds of reasons I have found the object-oriented computer language C++ a better medium than mathematics 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 [99, Appendix A].

[2]An example here is how 'hierarchical chunking' emerged as a powerful technique for modelling the learning of a first language (Section 4.6).
A simple example with learning syntactic structures relates to the problem that, while '`th`' is clearly a chunk in the early stages of learning, because it is the most frequently occurring pair of neighbouring letters, it is not a chunk at later stages when the MK10 program is trying to find structure in samples like this: '`...whathewanted...`'. The problem can be overcome by re-parsing the text at every stage, something that is easy to express with a program but more awkward to do with mathematics.

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 [8] and now including many mainstream computer languages including C++, was driven mainly by the realisation that software is 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 mainly online, so that readers may have the convenience of live links, and so that computer-powered searching reduces the need for an index. But an index is provided for readers who wish to read a printed copy of the book.

A summary of the main ideas in the book is in Appendix B.4.

Unique selling points for the SPTI are described in Chapter 2.

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

Some key terms are defined in Appendix A.

There is a note on copyright in Section 1.5.

Potential risks of the development of super-intelligent AI are discussed in Appendix C.

1.5 A note on copyright

Because this book is about the SPTI and the SPCM, much of it is drawn from existing peer-reviewed publications, mostly peer-reviewed open-access journals with CC-BY licences, and likewise for my previous book [86].

For this book, I have obtained the necessary permissions for the inclusion of figures and text for which I do not own the copyright.

A different issue relating to copyright is discussed in Appendix I.

2 Unique selling points

The SPTI has several unique selling points (USPs) but for the sake of clarity, the main USPs are summarised in Section 2.1, next, while the others follow in Section 2.2.

2.1 Main USPs

- *Intelligence via IC.* A central idea is that many aspects of intelligence, perhaps all aspects, may be understood as compression of information.

  The SPTI is, arguably, the most fully-developed theory of intelligence that recognises the importance of IC across *diverse* aspects of natural intelligence and, by conjecture, *all* aspects of intelligence.

- *Versatility via the SP-Multiple-Alignment concept.* More specifically, compression of information is achieved via the powerful concept of *SP-Multiple-Alignment*. This is largely responsible for the versatility of the SPTI (Chapters 7 and 9), the latter chapter including strengths across diverse aspects of intelligence (Section 9.1) and strengths in other areas too (Section 9.2).

  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 DNA for an understanding of biology. It may prove to be the 'double helix' of intelligence!*

- *Mathematics as IC.* A second **major discovery** is that *much of mathematics, perhaps all of it, may be understood as a set of techniques for compression of information, and their application.*

  This provides new foundations for mathematics which are radically different from any of the existing 'isms' in the foundations of mathematics (Chapter 10, [100]).

- *A New Mathematics.* There is potential for the integration of mathematics with the SPTI to create a *New Mathematics* with many potential advantages in science and beyond (Chapter 11).

2.2 Other USPs

- *Advantages of the SPTI compared with DNNs.* The popularity of DNNs makes them the main competitors for the SPTI. But the SPTI has several advantages compared with DNNs (Section 8.3 and [105]).

- *The SPTI and computing.* Apart from its strengths as a theory of AI, and its strengths in natural learning, perception and cognition, the SPTI has potential as a theory of computing, with substantial potential benefits (Section 11.2).

- *Seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, in any combination.* With respect to the development of AGI, an important strength of the SPTI is how it supports the seamless integration of diverse aspects of intelligence and diverse kinds of knowledge, in any combination (Section 8.2).

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

- *Transparency in the organisation and workings of the SPTI.* The SPCM provides transparency in the way it structures its knowledge, and there is an audit trail for all its workings [104]. These features, which contrast sharply with, for example, the lack of transparency in DNNs, are likely to prove useful in, for example, legal disputes about AI or in minimising the potential risks of AI.

- *Modest demands for data and for computational resources.* By contrast with the huge demands for data and for computational resources by DNNs, there is clear potential in the SPTI for much smaller demands for those things [105, Section 9].

- *No need for the 'theft' of the works of creative people.* By contrast with the building of knowledge required by such systems as ChatGPT, there is no need for linguistic or other works of people, including such creative people as Sir Elton John, Sir Paul McCartney, and so on.

  The intelligence of the SPTI stems from the compression of information (Chapters 4 and 5), it is not borrowed from the intelligence of people.

  Naturally, learning the syntax and semantics of a language requires exposure to that language. But, for example, finding the shortest route between two places requires the intelligent analysis of a map. It is unlikely to be solved by finding an appropriate pattern of words within, for example, a ChatGPT database.

- *Generalisation, over-generalisation, and under-generalisation.* The SPTI conceptual framework suggests that remarkably simple principles govern

the phenomena of generalisation, and the correction of over- and under-generalisations (Section 6.4.5). This analysis is indebted to Solomonoff's APT (Appendix D).

- *In unsupervised learning, reducing or eliminating the corrupting effect of errors in data.* How, in unsupervised learning, the SPTI may reduce or eliminate the corrupting effect of 'dirty data' is described in Section 6.4.6. This analysis appears to be unique to the SPTI.

- *How to learn usable knowledge from a single exposure or experience.* Like people, the SPTI can learn usable knowledge from a single exposure or experience [105, Section 7]. This contrasts sharply with the large quantities of data and large computational resources that are needed by DNNs before data is ready for use.

3 In the quest for artificial general intelligence, the importance of research strategy

As its title suggests, this section is about how, in the quest for AGI, we may benefit from a top-down, research strategy, and we would be handicapped by a bottom-up depth-first strategy.

The latter type of research strategy, which is almost universal in AI research, concentrates research efforts on one small aspect of intelligence, with the implicit expectation that it will be possible to generalise via several such studies, and thus reach AGI. Few projects get beyond one or two micro-theories, each of one small area, and none have yet yielded a satisfactory general theory of AGI.

Some of the problems associated with bottom-up, depth-first research strategies are described in Section 3.1, next, and the benefits of a top-down strategy are described in the following Section 3.2.

3.1 Problems with a depth-first strategy in the development of theory

As described in subsections below, various authors have drawn attention to the problem of fragmentation and related issues in AI 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' [51]. In that paper, he exhorted researchers to tackle "a genuine slab of human behaviour" [51, 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* [52] and a programme of research developing the Soar cognitive architecture [42, 52], aiming for a unified theory of cognition.

3.1.2 Pamela McCorduck and fragmentation of research

Again, in connection with the fragmentation in AI, science writer Pamela McCorduck writes:

> "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." [49, p. 417].

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." [49, 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." [49, 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 small areas within 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." [40, location 1004].

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

The main 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." [51, 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.*

3.1.5 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 seems 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 suggests that the assumption is a bit like climbing a tree in the belief that this would somehow enable one to reach the moon [23, p. 119].

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

3.1.6 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." [47, pp. 12–14] (emphasis in the original)

3.2 The benefits of a top-down research strategy

The quotes in Sections 3.1 and 3.1.4 are, in effect, calls for a top-down 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).
   - As noted in Appendix B, 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 may 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. Thus, 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 3.1.4, any micro-theory or theories that are 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 [25, p. 234].

   But as noted in Section 3.1.4, much research in AI has been, and, to a large extent, still is 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 any one group rarely gets beyond two areas, and so, when two or more groups are working on varied topics, there is much fragmentation of research.

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

The overarching goal of the SP research has been and still is to simplify and integrate observations and concepts across 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 foundation for that strategy is evidence that much of the workings of brains and nervous systems may be understood as IC (Chapter 4).

Another clue was provided by the bioinformatics concept of multiple sequence alignment (MSA, Section 5.3) which seemed to have the potential for the desired simplification and integration of concepts across a wide area.

As described in Section 6.3, the concept of MSA led to the development of the concept of *SP-Multiple-Alignment*. Despite its similarity with the concept of MSA, 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 previously-noted 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." (original emphasis) [47, p. 15].

4 Information compression in human learning, perception, and cognition

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.

This chapter describes some of the evidence for the importance of IC in the workings of brains and nervous systems, and in intelligence in people and other animals, drawing on [99], where further information may be found.

While it is clear from the evidence described in this chapter, itself drawing on [99], that IC is important in people and other animals, it is also clear that redundancy in information has several benefits, as described in Appendix E. Although it sounds paradoxical, information in people and other animals may contain

redundancy and be compressed at the same time. Much the same may be true of AI systems.

4.1 Pioneering research by Fred Attneave, Horace Barlow, and Satosi Watanabe

Claude Shannon's 'communication theory' [63,64], now called 'information theory', provided an inspiration for early writings by Fred Attneave [1, 2], Horace Barlow [3, 4] and Satosi Watanabe [74, 75] describing how IC may be seen in the workings of brains and nervous systems, which may themselves be seen as the foundation of human-level intelligence.

This idea has been investigated by various researchers up to the present (e.g. [14, 15, 35]), much of it within the framework of AIT (Section 1.3.1).

As noted above and described below, Horace Barlow has made major contributions to evidence and thinking about the importance of IC in intelligence in people and other animals. But I'm sorry to say that, after major contributions in this area, he changed his views on this topic. Appendix J, reproduced with permission from [99, Appendix B], summarises his revised views, together with comments from me, arguing that Barlow's revised views are largely wrong.

4.1.1 Some informational aspects of visual perception

In a paper called 'Some informational aspects of visual perception', Fred Attneave [1] 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)." [1, p. 184].

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." [1, p. 185].

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

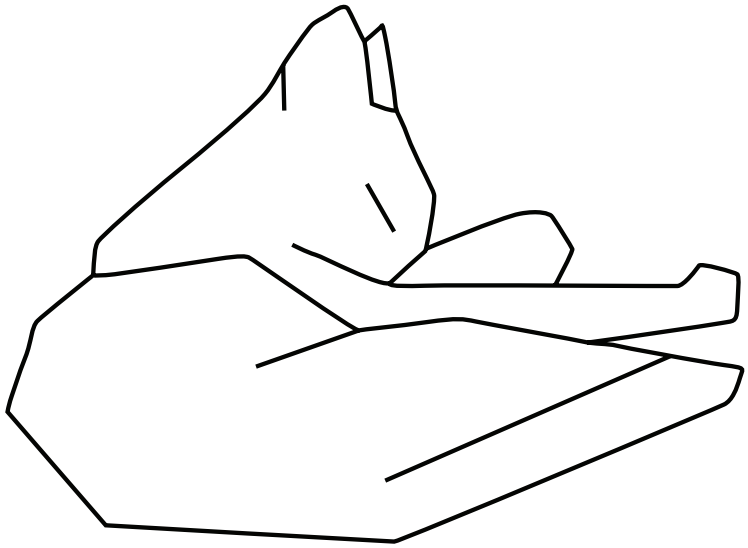

Figure 1: Drawing made by abstracting 38 points of maximum curvature from the contours of a sleeping cat, and connecting these points appropriately with a straight edge. Reproduced from Figure 3 in [1], with permission.

4.1.2 The need for compression of sensory information

In [3, p. 537], Barlow argues, on the strength of the large amounts of sensory information being fed, via the eyes, 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 that visual information, that:

> "... the storage and utilization of this enormous sensory inflow would be made easier if the redundancy of the incoming messages was reduced." [3, Summary].[3]

The phenomenon of 'lateral inhibition' in the retina ([31], [3, Section 5]) confirms that there is indeed a process that compresses visual information, much as Barlow suggested.

4.1.3 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 [3]. Later, he wrote:

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

> "... 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." [4, p. 210],

where 'find[ing] a less redundant code' means '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, some of it described in Chapter 4, drawing on the paper [99].

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 9.1 and 9.2).

4.2 Storage and transmission of information

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

- By allowing the creature 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 to (dangerous or advantageous) stimuli 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.
- Of course, small differences between views from our left and right eyes are interpreted by our brain as the distance from our eyes to the things that we are looking at (see footnote in Section 4.3).

4.3 Merging multiple views to make one

If, when we are looking at a landscape, we close our eyes for a moment and open them again, what do we see? Normally, it is the same as what we saw before, usually with small differences due to the differing positions of our left and right eyes.[4] But creating a single view out of the before and after views, means unifying the two patterns to make one and thus compressing the information, as shown schematically in Figure 2.

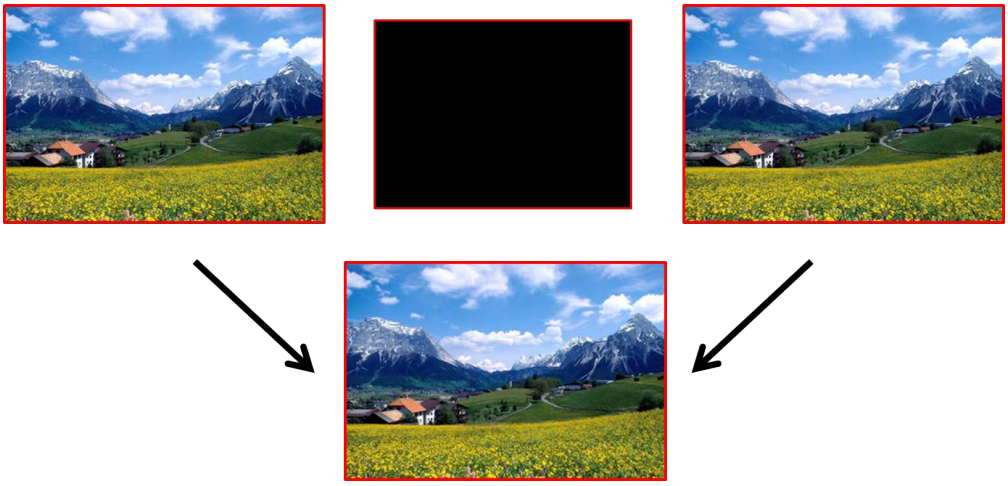

Figure 2: A schematic view of how, if we close our eyes for a moment and open them again, we normally merge the before and after views to make one. The landscape here is from Wallpapers Buzz (www.wallpapersbuzz.com), reproduced with permission.

It seems so simple and obvious that if we are looking at a landscape like the one in the figure, there is just one landscape even though we may look at it two, three, or more times. But if we did not unify successive views we would be like an old-style cine camera that simply records a sequence of frames, without any kind of analysis or understanding that, very often, successive frames are identical or nearly so.

There is more discussion of this kind of ICMUP processing in Appendix G.

4.4 Binocular vision

IC may also be seen at work in binocular vision:

[4]Differences that we interpret, as noted at the end of Section 4.2, as the distances from us of the things in the landscape that we are viewing.

> "In an animal in which the visual fields of the two eyes overlap extensively, as in the cat, monkey, and man, one obvious type of redundancy in the messages reaching the brain is the very nearly exact reduplication of one eye's message by the other eye." [4, p. 213].

In viewing a scene with two eyes, we normally see one view and not two. This suggests that there is a unification of views from the two eyes, with a corresponding compression of information.

A sceptic might say, somewhat implausibly, that the one view that we see comes from only one eye. But that sceptical view is undermined by the fact that, normally, the one view gives us a vivid impression of depth that comes from merging the two slightly different views from both eyes.

Strong evidence that, in stereoscopic vision, we do indeed merge the views from both eyes, comes from a demonstration with 'random-dot stereograms', as described in [90, Section 5.1] and with more detail in [39].

In brief, each of the two images shown in Figure 3 is a random array of black and white pixels, with no discernable structure, but they are related to each other as shown in Figure 4: both images are the same except that a square area near the middle of the left image is further to the left in the right image.

When the images in Figure 3 are viewed with a stereoscope, projecting the left image to the left eye and the right image to the right eye, the central square appears gradually as a discrete object suspended above the background.

Although this illustrates depth perception in stereoscopic vision—a subject of some interest in its own right—the main interest here is on how we see the central square as a discrete object. There is no such object in either of the two images individually. *It exists purely in the* relationship *between the two images, and seeing it means matching one image with the other and unifying the parts which are the same.*

4.5 IC and concepts of prediction and probability

In addition to empirical evidence for the role of IC in the storage or transmission of information, IC is closely related to concepts of prediction and 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 G): 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.

Figure 3: A random-dot stereogram from [39, Figure 2.4-1], reproduced with permission of Alcatel-Lucent/Bell Labs.

1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	Y	*A*	*A*	*B*	*B*	0	0
1	1	1	X	*B*	*A*	*B*	*A*	0	1
0	0	1	X	*A*	*A*	*B*	*A*	1	0
1	1	1	Y	*B*	*B*	*A*	*B*	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	*A*	*A*	*B*	*B*	X	0	0
1	1	1	*B*	*A*	*B*	*A*	Y	0	1
0	0	1	*A*	*A*	*B*	*A*	Y	1	0
1	1	1	*B*	*B*	*A*	*B*	X	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

Figure 4: Diagram to show the relationship between the left and right images in Figure 3. Reproduced from [39, Figure 2.4-3], with permission of Alcatel-Lucent/Bell Labs.

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 11.1.

The way in which the SPTI calculates absolute and relative probabilities for each SP-Multiple-Alignment is described in Section 6.3.12.

4.6 Research on the learning of a first language

More by luck than judgement with respect to the importance of IC within human learning, perception, and cognition and AI, I began research some time ago creating computer models of aspects of the learning of a first language by young children. Both phases of that research, described in [80], and outlined in the next two subsections, yield strong evidence for the importance of IC in those two aspects of learning.

4.6.1 The segmentation problem in the learning of a first language

Early work trying to understand how young children learn a first language aimed to understand how young children may discover the segmental structure of language (words and phrases), despite the fact that, for words at least, there appear to be no consistent physical markers for the beginnings and ends of words.

That last point is illustrated in Figure 5, showing the sound spectrogram for the spoken phrase 'On our website'. People normally speak in 'ribbons' of sound, without gaps between words or other consistent markers of the boundaries between words. In the figure, it is not obvious where the word 'on' ends and the word 'our' begins, and likewise for the words 'our' and 'website'. Just to confuse matters, there are three places within the word 'website' that look as if they might be word boundaries.

Computer models were created to see whether it was possible to discover word segments by heuristic search (see Appendix F) with alphabetic 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

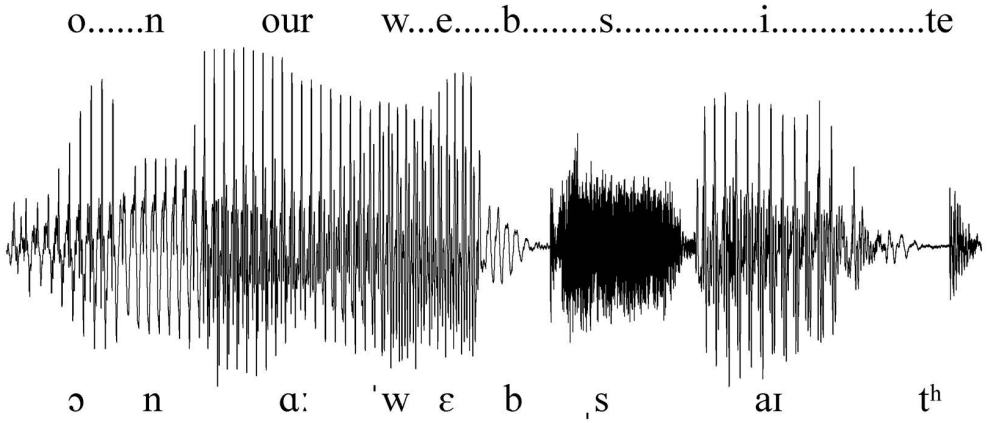

Figure 5: Waveform for the spoken phrase 'On our website' with an alphabetic transcription above the waveform and a phonetic transcription below it. With thanks to Sidney Wood of SWPhonetics (swphonetics.com) for the figure and for permission to reproduce it.

pairs of neighbouring elements.

That measure translates fairly directly into a measure of the IC that may be achieved via the Chunking-with-Codes technique (Appendix G.2.2), 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 6, 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.* [29]).

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

Figure 6: 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 [80], with permission.

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[5] Statistically significant results for the discovery of phrase structure in unsegmented text were obtained with the MK10 program [77].

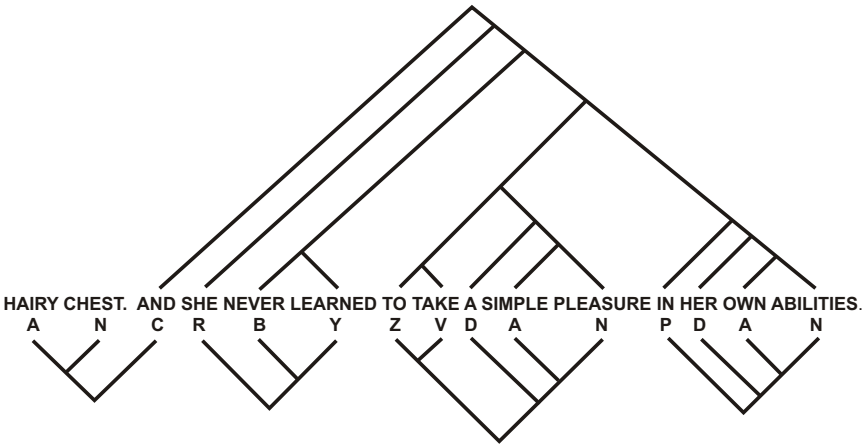

Figure 7: Phrase structure for one sentence processed with the MK10 program as described in the text. The 'correct' parsing, via human judgement, is shown above the sentence, and the best parsing by the MK10 program is shown beneath the sentence. Reproduced from Figure 1 (sentence 10) in [77], with permission.

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.

4.6.2 Discovering the grammars of English-like artificial languages

In later research, computer models were designed for the 'unsupervised' learning of the syntax 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 F).

As with the segmentation problem (Section 4.6.1), heuristic search, just mentioned, is normally needed in the unsupervised learning of SP-Grammars (Section 6.4). And, as with the segmentation problem, the best results by far were obtained

[5]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 7.

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, perception, and cognition.

Some examples of what has been achieved with the the 'SNPR' program for learning the syntax of English-like text with all spaces and punctuation removed may be found in [78].

4.6.3 Unsupervised learning

An important point about the MK10 program (Section 4.6.1) and the SNPR program (Section 4.6.2) is that they both embody a model of learning that is *unsupervised* in the sense that they do not depend on rewards or punishments or anything equivalent that may determine what is 'right' or 'wrong'.

This accords with varied evidence for the importance of unsupervised learning in people and other animals, as described in Section 6.4.

4.7 Barlow's change of view

In several papers, over a period of years, Barlow developed the ideas outlined in Sections 4.1.2 and 4.1.3. 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." [5, 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 [99, 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 for many years, and are continuing to be a source of insights about AI and cognition.

Notwithstanding Barlow's change of view, Chapter 4 and, in more detail, the paper [99] describe substantial evidence for the importance of IC in human learning, perception, and cognition.

5 The origin and development of the SP Theory of Intelligence

At all stages in the development of the SPTI and SPCM, the central principle has been, and continues to be the compression of information.

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

5.1 Early ideas

The importance of IC in human cognition became clear from my research on the learning of a first language (Sections 1.2 and 4.6).

Another inspiration arose from browsing the book *Introduction to Theoretical Linguistics* by John Lyons [45]. 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 1.2) 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, perception, 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 (Chapter 3).

The first step was the creation of a program for finding good full and partial matches between two sequences of atomic SP-Symbols. This is described in [86, Appendix A]. The reasons for this choice were:

- *Information compression*:

  - Pioneering research by Fred Attneave [1, 2], Horace Barlow [3, 4], and Satosi Watanabe [74, 75] 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 [99].
  - It seemed likely that 'good' full and partial matches between SP-Patterns (sequences of SP-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 SP-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 SP-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 The potential of multiple sequence alignments (MSAs) or something like them

Having developed a program for finding good full and partial matches between two sequences of SP-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 SP-Symbols. This kind of analysis, the previously-mentioned MSA, 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 SP-Symbols that match each other from one sequence to another. The number of such matches provides a measure of 'goodness' for each MSA.

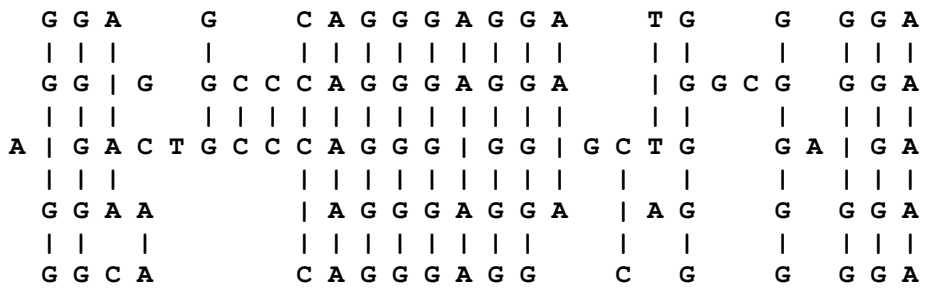

Figure 8: A good MSA amongst five DNA sequences. Reproduced from [86, Figure 3.1].

An example of a good MSA of five DNA sequences is shown in Figure 8.

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

To illustrate how the MSA concept, or something like it, may be harnessed to AI, two rather rough examples are shown in Figures 9 and 10.

```
man             mortal  (``All men are mortal'')
 |
man Socrates            (``Socrates is a man'')
       |
    Socrates    mortal  (``Therefore, Socrates is mortal'')
```

Figure 9: A simple example showing how some variation of the MSA concept might model an aspect of AI. Here, a familiar example of deductive reasoning is modelled very roughly via an MSA.

```
sheep wool warm-blooded has-legs bleats ... eats-grass milk-for-young
               |          |                                  |
mammal     warm-blooded has-legs talks ... has-mouth   milk-for-young
```

Figure 10: Another simple example showing how some variation of the MSA concept might model an aspect of AI. In this case, recognition of a sheep as a mammal is modelled very roughly via an MSA.

Chapter 7 has several much better examples showing how the SP-Multiple-Alignments concept from the the SPCM may model diverse aspects of intelligence.

5.4 Developing the SP-Multiple-Alignment concept

The concept of MSA was structured well enough to suggest how it might be developed in studies of human cognition or AI, but it was far from being usable in

the embryonic SPTI.

To create the new SP-Multiple-Alignment concept from the MSA 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 in 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 further ideas for how the system might be made to work, and any shortcomings 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. As noted earlier, 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 Chapter outlines 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. More detail may be found in [86].

6.1 High level view of the SPTI

The SPTI is conceived as a brain-like system as shown in Figure 11, 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 in the brain as *Old* information (red).

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.

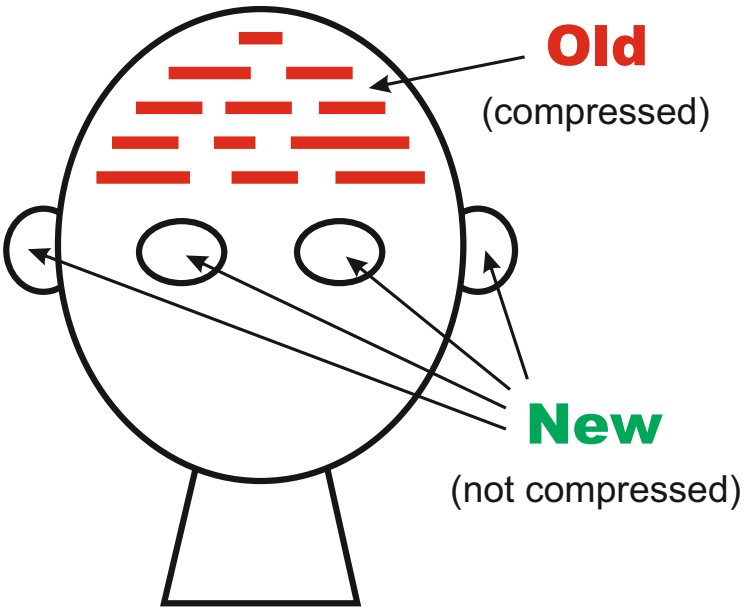

Figure 11: Schematic representation of the SPTI. Reproduced from Figure 1 in [89].

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 SP-Symbol '+' in arithmetic, or 'multiply' for the SP-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.

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 intelligence and diverse kinds of intelligence-related knowledge, in any combination (Section 8.2).

Examples of SP-Patterns may be seen in Figure 12, 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.

The addition of 2D SP-Patterns should open up the system for the representation and processing of diagrams and pictures. And 2D SP-Patterns may also serve in the representation of structures in three dimensions, as outlined in Section 6.4.7 and [90, Sections 6.1 and 6.2)].

2D SP-Patterns may also have a role in the representation and processing of parallel streams of information, as described in [91, 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.

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 13 (Section 6.3) contains the ID SP-Symbols '`N`', '`Ns`', '`4`', and '`#N`' at the left and right 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, so the same SP-Symbol 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 [86, Section 3.4] and [89, Section 4].

6.3.1 SP-Multiple-Alignment, Simplicity and Power

The SP-Multiple-Alignment concept is largely responsible for the intelligence-related and other strengths of the SPTI, described with examples in Chapter 7 and summarised in Chapter 9.

Although it is far from being trivially simple, the SP-Multiple-Alignment concept is remarkably simple compared with its versatility. Thus the relative Simplicity of the SPTI combined with its high descriptive and explanatory Power, is largely due to the high ratio of Simplicity to Power of the SP-Multiple-Alignment concept and the SPTI (Appendix B).

Because the scope of the SP-Multiple-Alignment concept is so large, the importance of that high ratio of Simplicity to Power is more significant than if the

the scope of the SP-Multiple-Alignment concept had been smaller (Appendix B.1).

As noted amongst the main USPs (Section 2.1), *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 MSA in bioinformatics.

An example of an SP-Multiple-Alignment is shown in Figure 12.

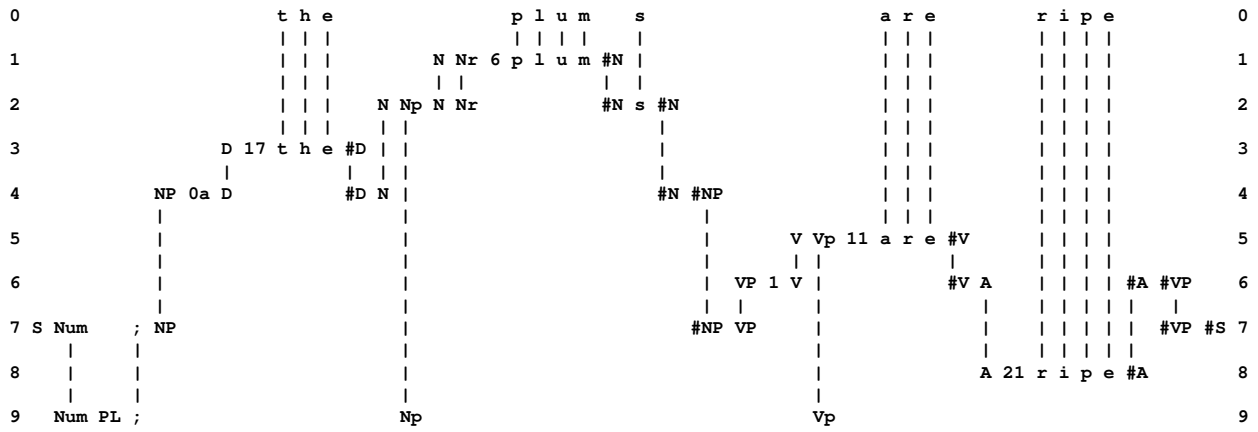

Figure 12: The best SP-Multiple-Alignment amongst several created by the SPCM that achieves the effect of parsing a sentence ('t h e p l u m s a r e r i p e') into its parts and sub-parts, as described in the text. The sentence in row 0 ia a New SP-Pattern, while each of the rows 1 to 9 contains a single Old SP-Pattern, drawn from a relatively-large repository of Old SP-Patterns. Reproduced from Figure A2 in [84].

The main components of the SP-Multiple-Alignment concept, illustrated in Figure 12, are these:

- Row 0 contains one New SP-Pattern representing information that has been received recently from the system's environment (Section 6.1). In this example, the SP-Pattern in row 0 is a sentence.

  In other examples, row 0 may sometimes contain more than one New SP-Pattern. And in other examples, the New SP-Pattern(s) may represent other kinds 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 13, and each Old SP-Pattern in the SP-Multiple-Alignment and in the SP-Grammar represents a grammatical structure, where that category includs words.

  More generally, as noted above, Old SP-Patterns may represent many other kinds 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 (See, for example, Figure 31, 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 13: In this SP-Grammar, each SP-Pattern starts and ends with one or more SP-Symbols which are called 'ID' SP-Symbols representing a grammatical category. The character '!' in the SP-Grammar serves to mark an SP-Symbol as being an 'ID SP-Symbol' and, in each SP-Pattern, the unmarked SP-Symbols are 'C' or 'contents' SP-Symbols (see Section 6.2). As noted in Section 6.2.2, the character '!' only serves as a marker of ID SP-Symbols in some contexts and not in others.

6.3.3 How SP-Multiple-Alignments are built up

Here is a summary of how an SP-Multiple-Alignments like the one shown in Figure 12 is built up:

1. At the beginning of processing, the SPCM has the afore-mentioned store of Old SP-Patterns, as shown in Figure 13.

   When the SPCM is more fully developed, those Old SP-Patterns would have been learned from raw data as outlined in Section 6.4, 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 12.

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 the ID SP-Symbols in Old SP-Pattern(s) with which it has been unified.

4. At all stages, the search for good full and partial matches between SP-Patterns is achieved via ICMUP (Appendix G), which is itself based on the creation of hit structures as described in Appendix D.3.

5. 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.

6. In later stages, the program searches for matches between SP-Multiple-Alignments established in earlier stages. Figures 14 and 15 shows the order of pairwise alignments in the creation of the SP-Multiple-Alignment in Figure 12. In any pairing where one or both of the pairs is itself an SP-Multiple-Alignment, that SP-Multiple-Alignment is represented by the main SP-Pattern within the SP-Multiple-Alignment.

7. In SP-Multiple-Alignments, IC may be achieved by collapsing the whole SP-Multiple-Alignment into a single sequence of SP-Symbols and thus unifying matching SP-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.

8. Details of how IC for any one full or partial SP-Multiple-Alignment is calculated are given in Section 6.3.6.

9. As processing proceeds, similar pair-wise matches and unifications eventually lead to the creation of SP-Multiple-Alignments like that shown in Figure 12.

   At every stage, all the SP-Multiple-Alignments that have been created are evaluated in terms of IC, and, at each stage, 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 12.

10. This process of searching for good SP-Multiple-Alignments in stages, with a selection of good partial solutions at each stage, is an example of heuristic search, as described in Appendix F.

    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 with a reasonable computational complexity, but it cannot normally 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 16, in which the SP=Pattern '`NP D #D N #N #NP`' appears twice.

    Another example, where a given SP=Pattern (X L1 #L1 X #X L2 #L2 #X) is repeated with an SP-Multiple-Alignment, may be seen in Figure 27 in Section 7.1.

```
           p l u m
           | | | |
ID32 Nrt 6 p l u m #Nrt

           r i p e
           | | | |
ID33 A 21  r i p e #A

           t h e
           | | |
ID38 D 17  t h e #D

              a r e
              | | |
ID41 V Vpl 11 a r e #V

                   s
                   |
ID70 N Np N Nr #N  s #N

ID90       N Nr 6 p l u m #N
           | |            |
     N Np N Nr            #N s #N

           D 17 t h e #D
           |           |
ID99 NP 0a D           #D N #N #NP

           V Vp 11 a r e #V
           |             |                          |
ID101 VP 1 V             #V A #A #VP
```

Figure 14: A sequence of alignments, created in the process of constructing the SP-Multiple-Alignment shown in Figure 12. The sequence continues in Figure 15. In both figures, any pairing where one or both of the pairs is itself an SP-Multiple-Alignment, that SP-Multiple-Alignment is represented by the main SP-Pattern within the SP-Multiple-Alignment.

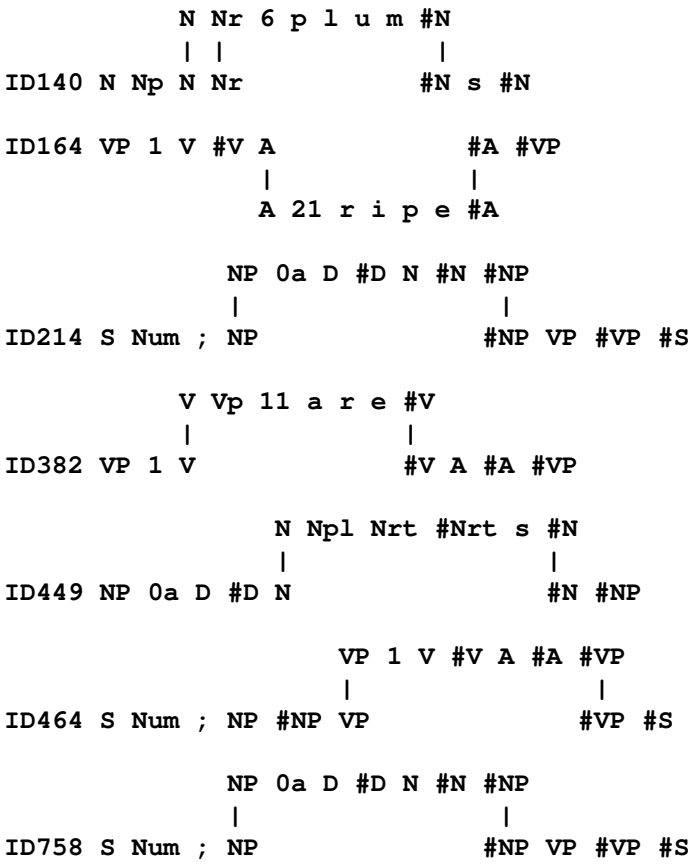

Figure 15: A continuation of the sequence of alignments in building the SP-Multiple-Alignment shown in Figure 12, as described in the caption to Figure 14. As before, any pairing where one or both of the pairs is itself an SP-Multiple-Alignment, that SP-Multiple-Alignment is represented by the main SP-Pattern within the SP-Multiple-Alignment. For the whole sequence shown in Figures 14 and 15, notice that the SP-Pattern '`Num PL ; Np Vp`' in row 9 of Figure 12 and its alignment with other SP-Patterns in the figure has been omitted from the sequence. This is because of the relative complexity of its alignments. These are discussed in Section 6.3.4.

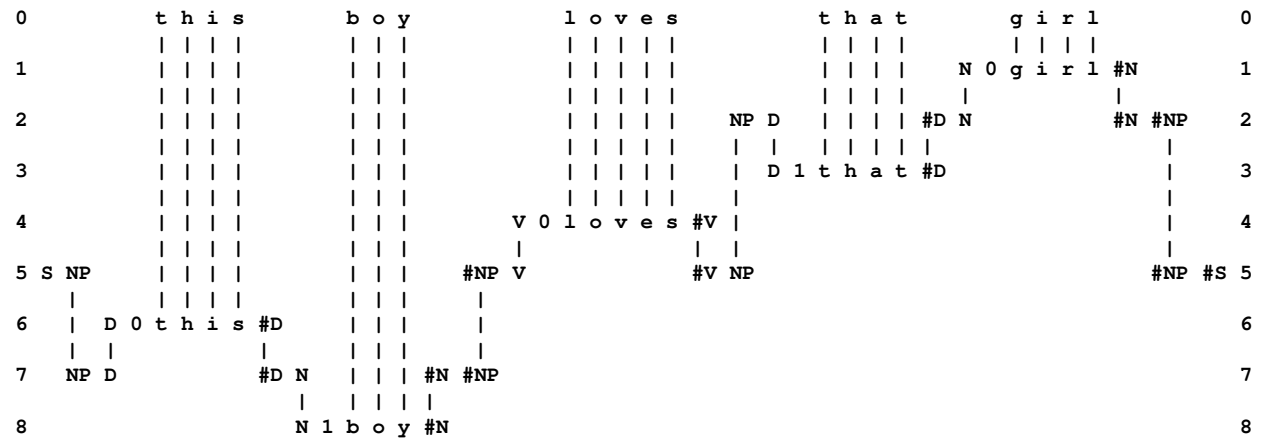

Figure 16: An SP-Multiple-Alignment showing how an Old SP-Pattern may appear more than once in the SP-Multiple-Alignment. In this case it is the SP-Pattern '`NP D #D N #N #NP`' which appears in row 7 and in row 2. Reproduced from [86, Figure 3.4 (a)].

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

Regarding the SP-Multiple-Alignment shown in Figure 12, an aspect not discussed so far is the role of the SP-Pattern '`Num PL ; Np Vp`' shown in row 9 of the figure.

Clues to the role of that SP-Pattern lie in the SP-Symbols '`PL`', '`Npl`', and '`Vpl`' within the SP-Pattern in row 9. The first of these SP-Symbols, '`PL`', indicates 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 9 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 and likewise for singular subjects and singular main verbs, which would be marked with the SP-Pattern '`!Num !SNG !; Ns Vs`'.

The plural and singular versions of the rule are the last two SP-Patterns in the SP-Grammar in Figure 13,

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 method in Prolog and other areas of computer science where variables are used as described in, for example, [10, Chapter 12].

The method illustrated in Figure 12 and described above and in [86, 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.

6.3.5 Versatility of the SP-Multiple-Alignment concept

As noted in the caption to Figure 12, 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 other AI-related areas and in areas without much connection with AI.*

Some of the versatility of the SP-Multiple-Alignment concept is described quite fully in subsections within Chapter 7, and the SPTI's versatility is summarised in Chapter 9, including AI-related versatility summarised in Section 9.1 and versatility in other areas summarised in Section 9.2).

As noted in Appendix G, 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 [106]. This generalisation is probably the main reason for the versatility of the SPTI outlined above.

6.3.6 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 [89, Section 4.1] and [86, Section 3.5].

6.3.7 Preliminaries: calculating the encoding costs of SP-Symbol types and SP-Patterns

Associated with each SP-Symbol type is a *code* or bit-pattern that serves as a measure of the 'cost' in bits of that SP-Symbol type as it appears in any SP-Pattern or SP-Multiple-Alignment that contains it.

The sizes of the codes are calculated (as described in Section 6.3.8, below) so that frequently-occurring ID SP-Symbols have shorter codes than ID SP-Symbols that occur more rarely.[6].

Given an SP-Multiple-Alignment like the one shown in Figure 12, one can derive a *code-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-pattern in the same order that they appear in the SP-Multiple-Alignment.

The code-pattern derived in this way from the SP-Multiple-Alignment shown in Figure 12 is 'S PL 0a 17 6 1 11 21 #S'. This is, in effect, a compressed representation of those SP-Symbols in the New SP-Pattern that form hits with Old SP-Symbols in the SP-Multiple-Alignment.

In this case, the code-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-pattern will represent only those New SP-Symbols that do form hits with Old SP-Symbols.

[6]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 a code-pattern derived in this way, we may calculate a 'compression difference' as:

$$CD = b_n - b_e, \tag{2}$$

or a 'compression ratio,' as:

$$CR = b_n / b_e, \tag{3}$$

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-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:

$$b_n = \sum_{i=1}^{h} c_i, \tag{4}$$

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

b_e is calculated as:

$$b_e = \sum_{i=1}^{s} c_i, \tag{5}$$

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

6.3.8 Encoding individual SP-Symbol types

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 type. But the SPCM uses variable-length codes for SP-Symbols, assigned in accordance with the Shannon-Fano-Elias coding scheme [21] so that the shortest codes represent the most frequent SP-Symbol types and *vice versa*.

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

$$f_{st} = \sum_{i=1}^{p} (f_i \times o_i) \tag{6}$$

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.

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.

6.3.9 Analysis of a sentence

With the example shown in Figure 12, 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, described in Section 6.3.7, for deriving an encoding from the SP-Multiple-Alignment in Figure 12, the result, as noted above, is the code-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-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, taking account of the size of the bit-pattern for each SP=Symbol.

6.3.10 Production of a sentence

As mentioned in Section 6.3.7, above, the way in which a code-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-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 17. All the words of the sentence '`t h e p l u m s a r e r i p e`' 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 9), '`0a`' picks out the SP-Pattern for noun phrases (in row 4), '`17`' picks out '`D 17 t h e #D`', and so on.

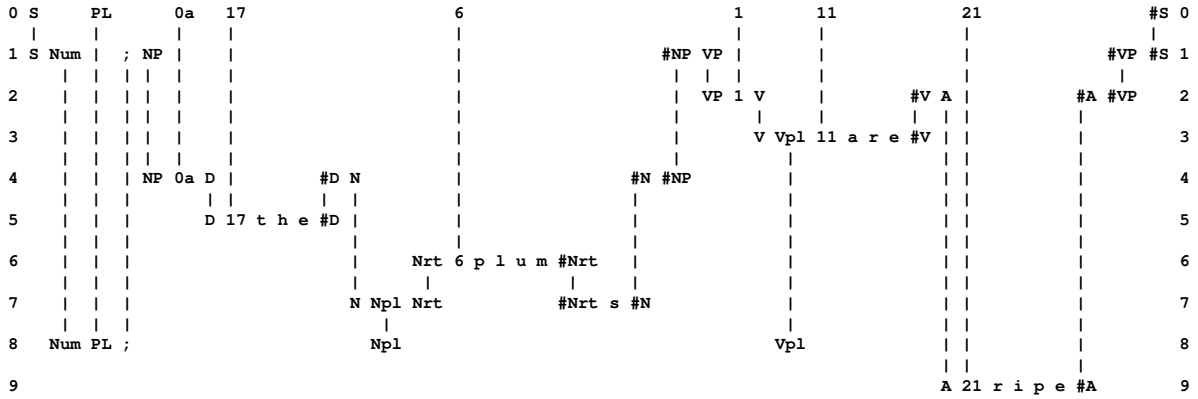

Figure 17: The best SP-Multiple-Alignment created by the SPCM that achieves the effect of decoding the code-pattern, '`S PL 0a 17 6 1 11 21 #S`', resulting in the recreation of the sentence '`t h e p l u m s a r e r i p e`', as described in the text. The SP-Pattern in row 0 ia a New SP-Pattern representing the encoding of the sentence, while each of the rows 1 to 9 contains a single Old SP-Pattern, and each such Old SP=Pattern contains one or more characters from the sentence '`t h e p l u m s a r e r i p e`'. So the SP-Multiple-Alignment has in effect recreated that original sentence.

6.3.11 How is it possible for the SPCM to recreate a sentence from a compressed version of that sentence?

What is described in Section 6.3.10 may seem like magic. How is it possible to recreate a compressed sentence using IC in exactly the same way as was done to create the compressed sentence?!

The answer is quite simple, without any mystery! In order to recreate a sentence from an encoding such as '`S PL 0a 17 6 1 11 21 #S`' representing a compressed version of the sentence, each SP-Symbol in the encoding is made 'fatter' by adding several bits to the bit-pattern for that SP-Symbol. Then with an artificially-fattened bit-pattern for each code SP-Symbol in the encoding, the SPCM may operate in the normal way by compressing that fattened bit-pattern.

6.3.12 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 [86, Section 3.7] and [89, Section 4.4].

The formation of SP-Multiple-Alignments in the SPTI supports several kinds of probabilistic reasoning, as described in Section 7.12.1. The core idea is that, within a given SP-Multiple-Alignment, any Old SP-Symbol, or group of Old SP-Symbols, 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.

6.3.13 Absolute probabilities

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

$$N = |A|^L. \tag{7}$$

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 L) is close to:

$$p_{ABS} = |A|^{-L}. \tag{8}$$

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

6.3.14 Is it reasonable to assume that an encoding derived from an SP-Multiple-Alignment is random or nearly so?

Why should we assume that the code derived from 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.

6.3.15 Relative probabilities

The absolute probabilities of SP-Multiple-Alignments, calculated as described in Section 6.3.13, 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 SP-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 SP-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 SP-Symbols from New which are encoded by that SP-Multiple-Alignment. We will call these SP-Symbols the *reference set of SP-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 SP-Symbols from New, neither more nor less.[7]

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 SP-Symbols appearing in a given row also appear in another row in the same order.[8] 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 SP-Multiple-Alignments:

$$p_{A\_SUM} = \sum_{i=1}^{i=R} p_{ABS_i} \tag{9}$$

   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.

5. For each SP-Multiple-Alignment in the reference set, calculate its relative probability as:

$$p_{REL_i} = p_{ABS_i} / p_{A\_SUM}. \tag{10}$$

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

$$p_{a\_SUM} = \sum_{i=1}^{i=r} p_{ABS_i} \tag{11}$$

   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

[7] 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 SP-Symbols *or any super-set of that set*. It is not clear at present which of those two definitions is to be preferred.

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

SP-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 SP-Symbols. It may happen that some other set of SP-Symbols from New is the focus of interest. In this case a different reference set of SP-Multiple-Alignments may be conceived and relative values for those SP-Multiple-Alignments may be calculated as described above.

6.3.16 Relative probabilities of SP-Patterns and SP-Symbols

It often happens that a given SP-Pattern from Old or a given alphabetic SP-Symbol type, within SP-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 SP-Pattern or alphabetic SP-Symbol type to be higher than if it appeared in only one SP-Multiple-Alignment. To take account of this kind of situation, the SPCM calculates relative probabilities for individual SP-Patterns and alphabetic SP-Symbol types in the following way:

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

2. For each SP-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 an SP-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 SP-Symbol types which appear anywhere in the SP-Patterns identified in step 2.

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

With regard to alphabetic SP-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.

6.3.17 Comparison of SP-Multiple-Alignments that do not encode the same SP-Symbols from New

It is true that, when we compare SP-Multiple-Alignments, we should compare like with like in terms of the SP-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 SP-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 SP-Symbols in New but this has not, so far, been investigated.

6.4 Unsupervised learning in people and other animals

The main focus of learning in the SPTI is on *unsupervised* learning, meaning learning without any of the kinds of assistance described in Gold's theory of language learning [29]: assistance from one or more 'teachers', the marking of the learner's utterances as 'right' or 'wrong', or the grading of data from simple to complex.

The main reasons for this focus on *unsupervised* learning are:

- Substantial evidence for the importance of unsupervised learning in the workings of brains and nervous systems [99].
-
- Since the MK10 program, described in Sections 4.6.1, and the SNPR program, described in 4.6.2, are designed to achieve IC in accordance with the evidence described in Section 4 and [99], and since IC is intrinsic to the data to be compressed, without any place for extrinsic values such as 'right' or 'wrong', the compression processes in both programs are intrinsically unsupervised.
- 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 in the first paragraph of this section, above. For example, Christy Brown [11] 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, which meant that, throughout his childhood, there was little or no possibility for people to correct his language.

- Evidence from everyday experience that, despite the existence of schools and colleges, most of human learning is unsupervised.
- The working hypothesis 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 8.3).

As with the creation of SP-Multiple-Alignments (Section 6.3), IC is central in unsupervised learning in the SPTI—and this is partly because: SP-Multiple-Alignments and their role in compressing information play a major part within unsupervised learning within the SPTI; and partly because IC is prominent in the remaining parts of unsupervised learning within the SPTI.

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

6.4.1 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 18 (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 18: (a) A simple SP-Multiple-Alignment from which, via the SPTI, Old SP-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 18 (b).

There are two important features of this learning:

- Each of the words derived from New patterns shown in Figure 18 (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 18 (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`'.

The example just described, captures the essentials of 'transfer learning' as described in Section 8.3, bullet point 8, and [105, Section 8].

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 F) 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 SP-Pattern in terms of **G**. Here, **E** is constructed as described in Section 6.3.6.

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

$$g = \sum_{i=1}^{i=g} (\sum_{j=1}^{j=k_i} s_j) \tag{12}$$

where k_i is the number of SP-Symbols in the ith SP-Pattern and s_j is the encoding cost of the jth SP-Symbol in that SP-Pattern.

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

6.4.2 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 8 of Figure 12. 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 ([105, 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 directly 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:

  > "... the first time you train a convolutional network you train it with thousands, possibly even millions of images of various categories." [25, p. 124].

- The model of language learning proposed by Gold [29], 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.

6.4.3 An outline of unsupervised learning in the SP70 version of the SPCM

Figure 19 provides an outline of unsupervised learning in the SP70 program, almost the same as SP71, the most complete version of the SPCM.

```
SP70()
{
     1 Read a set of patterns into New. Old is initially empty.
     2 Compile an alphabet of alphabetic symbol types in New and,
          for each type, find its frequency of occurrence and the
          number of bits required to encode it.
     3 While (there are unprocessed patterns in New)
     {
          3.1 Identify the first or next pattern from New as the
               'current pattern from New'.
          3.2 Apply the function CREATE-MULTIPLE-ALIGNMENTS() to
               create multiple alignments, each one between the
               current pattern from New and one or more patterns from Old.
          3.3 During 3.2, the current pattern from New is copied into Old,
               one symbol at a time, in such a way that the current pattern
               from New can be aligned with its copy but that any one
               symbol in the current pattern from New cannot be aligned
               with the corresponding symbol in the copy.
          3.4 Sort the multiple alignments formed by this function in order
               of their compression scores and select the best
               few for further processing.
          3.5 Process the selected multiple alignments with the function
               DERIVE-PATTERNS(). This function derives encoded
               patterns from multiple alignments and adds them to Old.
     }

     4 Apply the function SIFTING-AND-SORTING() to create one or
          more alternative grammars for the patterns in New, each
          one scored in terms of minimum length encoding principles.
          Each grammar is a subset of the patterns in Old.
}
```

Figure 19: The organisation of the SP70 version of the SPCM.

6.4.4 An example

When New contains the eight sentences shown in Figure 20, the best grammar found by the SPCM is the one shown in Figure 21.

```
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 20: Eight sentences supplied to the SPCM as eight New SP-Patterns.

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'.

```
< %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 21: The best SP-Grammar found by the SPCM when New contains the eight sentences shown in Figure 20.

In the pattern-generation phase of processing, the SPCM does form SP-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.

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

As Chomsky [18] 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 22. 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 6.4.6, children also have to cope with dirty data meaning 'wrong' utterances that are part of the sample from which they learn.

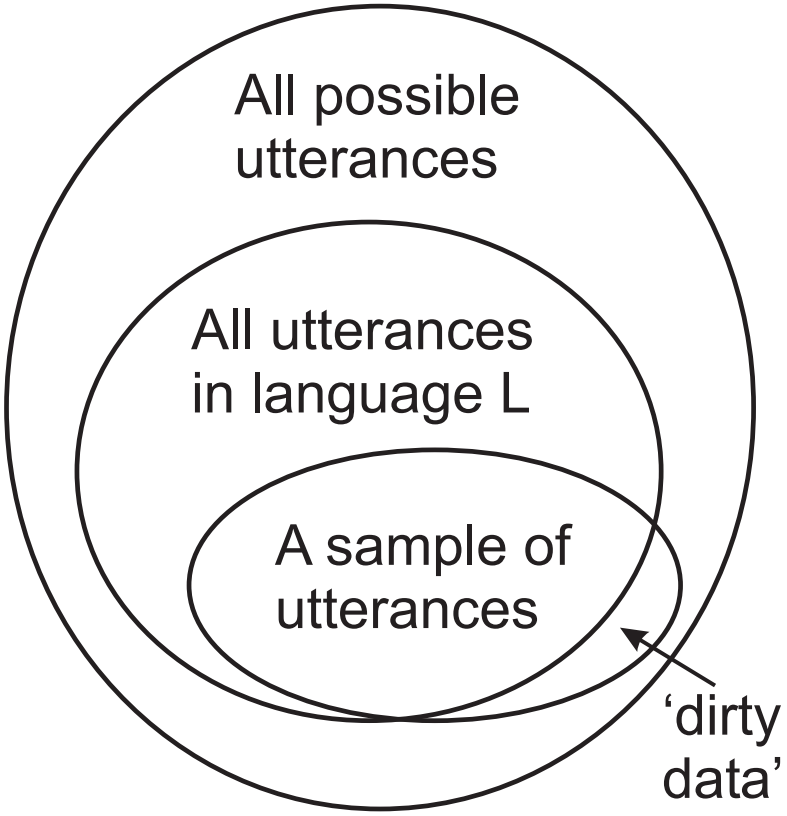

Figure 22: Categories of utterances involved in the learning of a first language, **L**. In ascending order size, they are: the finite sample of utterances from which a child learns; the (infinite) set of utterances in **L**; and the (infinite) set of all possible utterances. Adapted from Figure 7.1 in [80], with permission.

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 6.3.6.

- 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.

- In those terms, the general aim is to minimise $(g + e)$, where g is the size of the SP-Grammar **G** and e is the size of the encoding **E** of the New linguistic data which are the bases for learning.

6.4.6 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 22).

The answer appears to be this: minimise $(g + e)$ during learning and then discard **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 also include valid information that just happens to be rare in the particular **I** that is the basis for the analysis.

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

The proposed method of dealing with dirty data is not guaranteed to distinguish precisely between valid data and dirty data, in the same way that 'only report events that have been confirmed' is not guaranteed to distinguish precisely between valid reports and false reports. But in general the proposed method of dealing with dirty data will work better with large **Is** than small ones.

This can be seen in the learning of a native language: errors are greatest when a child is young and **I** is small, and errors decrease progressively towards adulthood, when **I** is very much larger. Of course, 'errors' that appear in **G** not **I**, are likely to be seen as dialect forms, not errors in that dialect.

6.4.7 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 [53, Section 9], where the fourth dimension is the time dimension in videos and the like.

The unsupervised learning of 2D SP-Patterns . Although the SPCM as it is now (with 1D SP-Patterns) may in principle be developed for the representation of 2D SP-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 4.1.1), and that, information 'is further concentrated at those points on a contour at which its direction changes most rapidly' [1, 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 [90, 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 23), 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 pictures. This kind of processing may be achieved via an SP-Multiple-Alignment that accommodates 2D SP-Patterns and has been processed to find redundancies within and between SP-Patterns.

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 size or shape is not important. And if we want more accurate information, we can inspect the object from more different angles, or supplement vision with touch.

The unsupervised learning of 4D SP-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 [53, Section 9.3]. And most videos

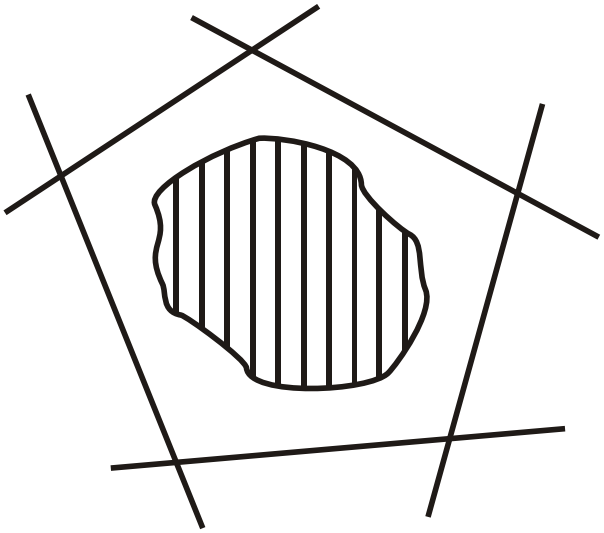

Figure 23: Plan view of a 3D object, with each of the five lines around it representing a view of the object, as seen from the side. Reproduced from [90, Figure 11].

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.

6.4.8 Plotting values for o, g, e and t

To provide a more rounded picture of unsupervised learning in the SPTI, Figure 24 shows cumulative values for critical variables as the 8 sentences shown in Figure 20 (Section 6.4.4) 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.

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.

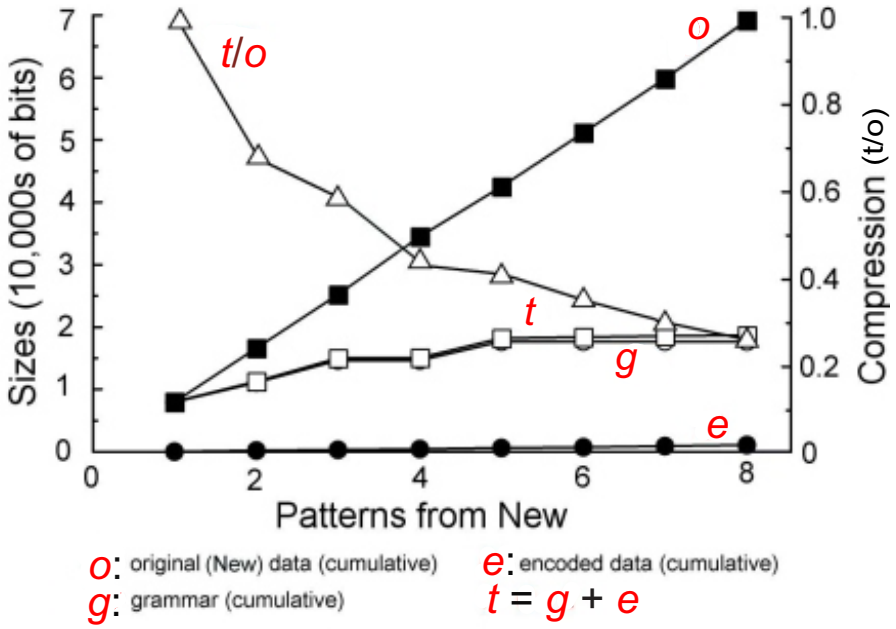

Figure 24: Changing values for the sizes of o, g, e and t and related variables as SPCM learning proceeds, with New SP-Patterns shown in Figure 20. Adapted from Figure 9.12 in [86].

6.4.9 Computational complexity

In common with other programs for unsupervised learning (and, indeed, other programs for finding good SP-Multiple-Alignments), the SPCM does not attempt to find theoretically ideal solutions.[9] This is because the abstract space of possible grammars (and the abstract space of possible SP-Multiple-Alignments) is, normally, too large to be searched exhaustively. In general, heuristic techniques must be used (Appendix F). 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 $O(\log_2 n \times nm)$ [83], where n is the size of the SP-Pattern from New (in bits) and m is the sum of the lengths of the SP-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 $O(m)$.

In the SPCM, the function is applied (twice) to the set of SP-Patterns in New so we need to take account of how many SP-Patterns there are in New. It seems reasonable to assume that the sizes of SP-Patterns in New are approximately constant.[10]

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 SP-Multiple-Alignments for any given SP-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 SP-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 $O(N)$

[9] This section is based on [86, Section 9.3.1]

[10] There is no requirement in the model that SP-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.

6.5 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 (Chapter 12).

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 [86, Sections 3.9, 3.10, and 9.2] and more briefly in Sections 6.3 and 6.4, above.

6.6 *SP-Neural*: a preliminary version of the SP Theory of Intelligence in terms of neurons and their interconnections 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 been developed in outline, with the concepts of SP-Pattern and SP-Multiple-Alignment expressed in terms of neurons and their interconnections and inter-communications ([94], [86, Chapter 11]).

The main challenge is how the processes of building SP-Multiple-Alignments, and of unsupervised learning, can be expressed in terms of neural processes.

6.6.1 Neural inhibition

In view of evidence for the importance of neural *inhibition* in the workings of brains and nervous systems [38], and in view of evidence for the importance of IC

in human learning, perception, and cognition [99], it seems possible that inhibition could be the neural basis for IC [94, 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 (Appendix G)*:

- Unification in the ICMUP concept is when (within a body of information **I**) two or more SP-Patterns that match each other are reduced to a single instance.
- Providing that the SP-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 at the same time. [31, 32].

With regard to lateral inhibition, Larry Squire and colleagues write:

> "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." [68, p. 578].

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 6.5), 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.

6.6.2 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 (see the last bullet point but two in Appendix B.4 and Chapter 6.6, above).

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.

6.7 Future developments and the SP Machine

In view of the potential of the SPTI in diverse areas (Chapter 9, the SPCM appears to hold promise as the foundation for the development of an industrial-strength *SP Machine*, described in [53], and illustrated schematically in Figure 25.

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 everuwhere. 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 '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 [105, 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.

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 [53]. 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, perhaps one of those provided by Nvidia. Each group of researchers or individual

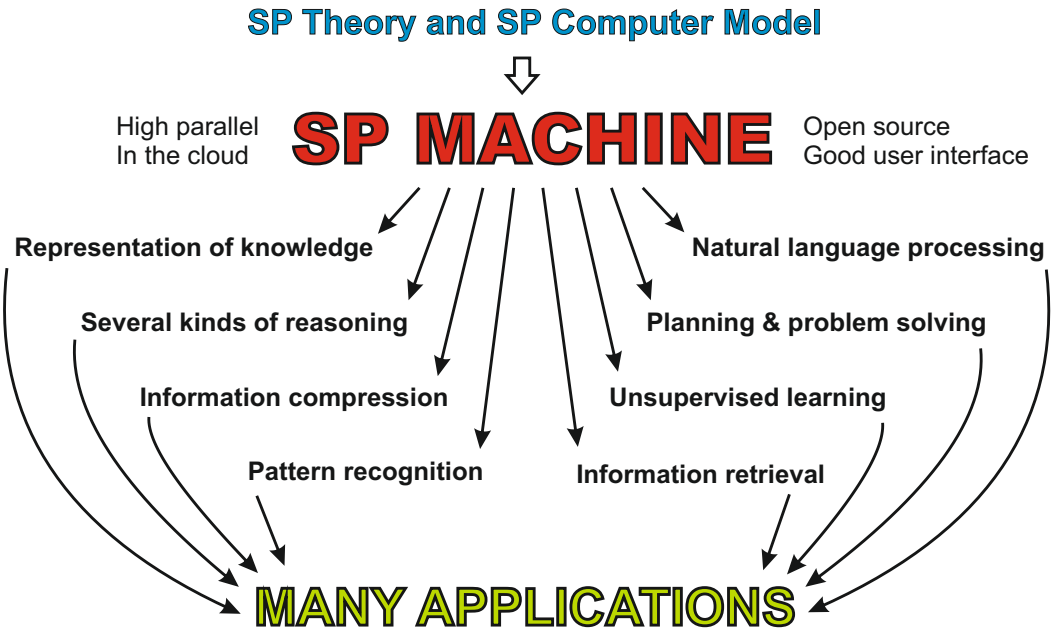

Figure 25: Schematic representation of the development and application of the SP Machine. Reproduced from Figure 2 in [89].

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

This chapter draws on examples in Chapters 5 to 9 in [86], with editing as appropriate.

As the title of this chapter suggests, the examples of SP-Multiple-Alignments show the kinds of things that may be done with the SPCM. The examples show some of the versatility of the SP-Multiple-Alignment concept, but SP-Multiple-Alignment concept is 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 is another example of recursive processing in Figure 65 in Section 11.2.1.

Regarding the recognition of a palindrome, Figure 27 shows the best SP-Multiple-Alignment produced by the SPCM with 'a c b a b a b c a' in New and SP-Patterns under the heading 'Old' in Figure 26. The SP-Multiple-Alignment may be seen as a recognition that the SP-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 26: SP-Patterns for processing by the SPCM to model the recognition of a palindrome.

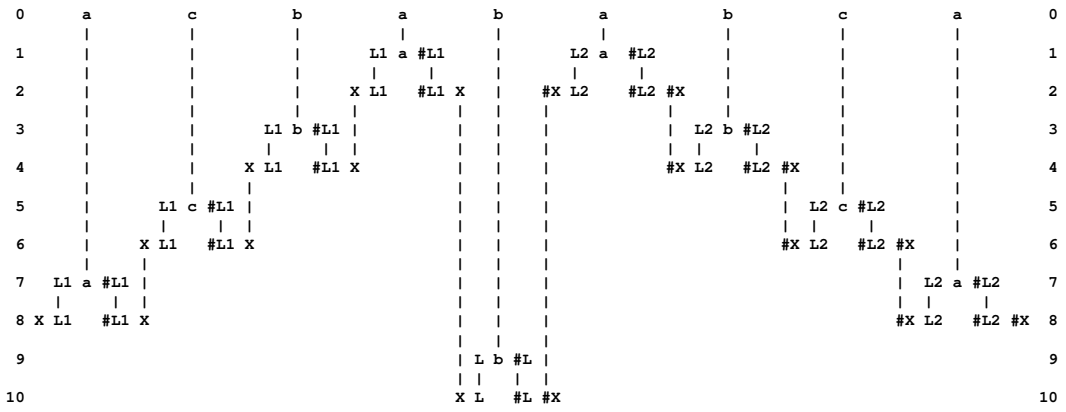

Figure 27: The best SP-Multiple-Alignment (in terms of compression) produced by the SPCM with the SP-Patterns from Figure 26.

7.2 Ambiguities in language

Natural languages are notoriously ambiguous, not only in their meanings but also in their syntax. An example showing syntactic ambiguity is the second sentence in 'Time flies like an arrow. Fruit flies like a banana', with uncertain origins.[11]

Figure 28 shows how the SPCM can accommodate the ambiguity of that second sentence, 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 SP-Multiple-Alignments formed for the same sentence.

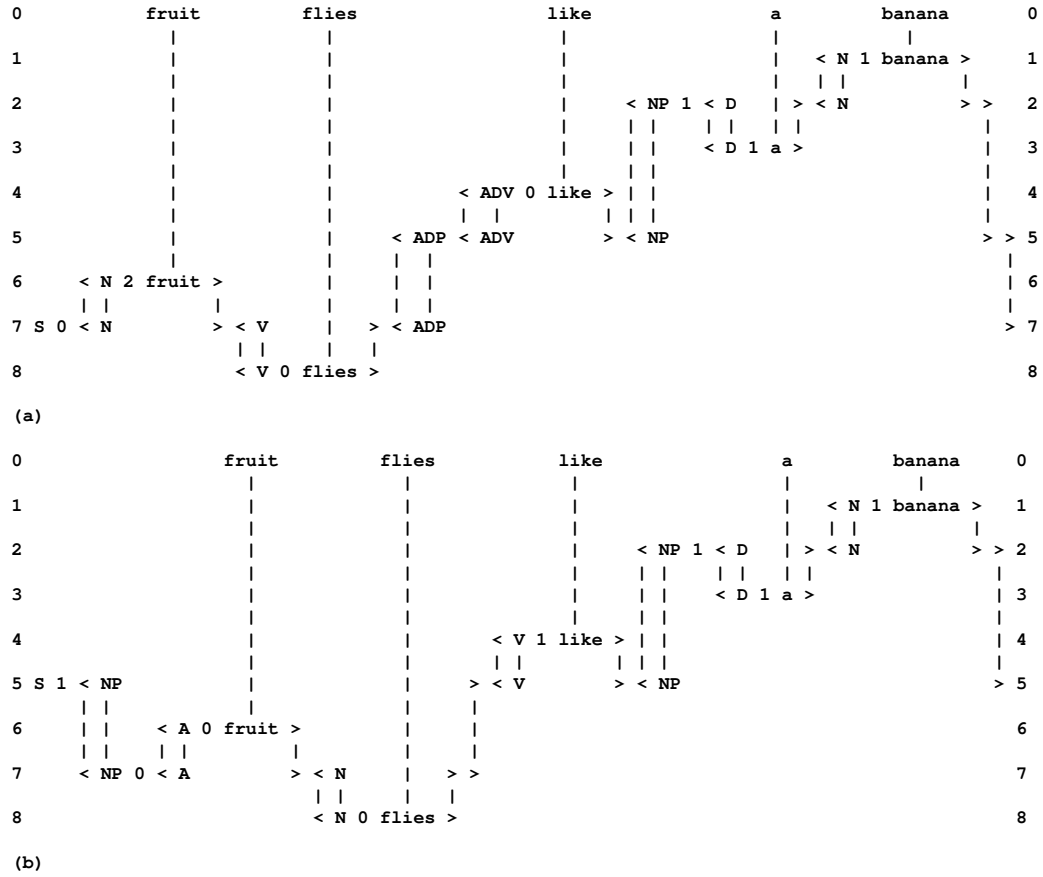

Figure 28: The two best SP-Multiple-Alignments found by the SPCM for the ambiguous sentence 'fruit flies like a banana' as the New SP-Pattern and with SP-Patterns in the repository of Old SP-Patterns representing grammatical rules including words. Reproduced from [86, Figure 5.1].

[11] Based on [86, Section 5.2.2].

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

Figure 29 (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 29 (b) shows how the SPCM achieves the same 'correct' parsing, except that the (b) sentence contains: an error of omission where the letter '`w`' is missing from the word '`t w o`'; an error of substitution where the letter '`m`' replaces the letter '`n`' in the word '`k i t t e n s`', and an error of addition where the letter '`x`' has been added to the word '`p l a y`'.[12]

In effect, the (b) 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`'.

Examples like this suggest that—by contrast with the way in which DNNs are liable to make large and unexpected errors in recognition ([105, Section 3], and bullet point 2 in Section 8.3)—the SPTI and its realisation in the SPCM, the process of parsing a sentence, have robust capabilities for recovering from errors in its data.

[12]This section is based on [89, Section 4.2.2].

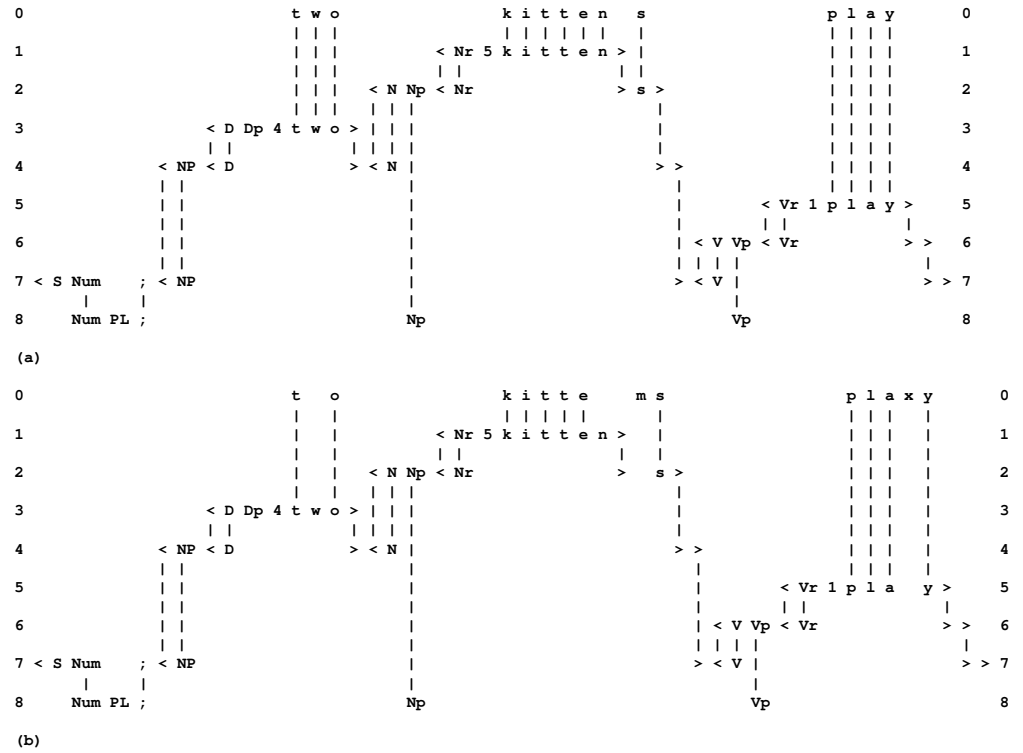

Figure 29: (a) The best SP-Multiple-Alignment created by the SPCM with a New SP-Pattern (representing a sentence to be parsed) shown in row 0, and a store of Old SP-Patterns like those in rows 1 to 8 (representing grammatical structures, including words); and (b) The best SP-Multiple-Alignment created by the SPCM with the same New SP-Pattern as in (a) but with errors of omission, substitution and commission as described in the text, and with the same set of Old SP-Patterns as before. (a) and (b) are reproduced from Figures 1 and 2 respectively in [87], with permission.

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.[13] 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 its 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 in this case ('P'), 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 its qualifying 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 grammarS (TG) [17] and, later, in systems like definite clause grammars [55], based on Prolog [27].

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 30.

[13] Based on [86, 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 30: A fragment of French SP-Grammar with SP-Patterns in the last 8 SP-Patterns for number dependencies (shown as 'SNG' and 'PL') and gender dependencies (shown a 'M' and 'F').

Apart from the use of SP-Patterns as the medium of expression, this SP-Grammar differs from systems like TG 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 30) do not contain any reference to number or gender. Instead, the SP-Grammar contains the eight SP-Patterns shown at the end of Figure 30, whose is function is described next.

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 30 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.

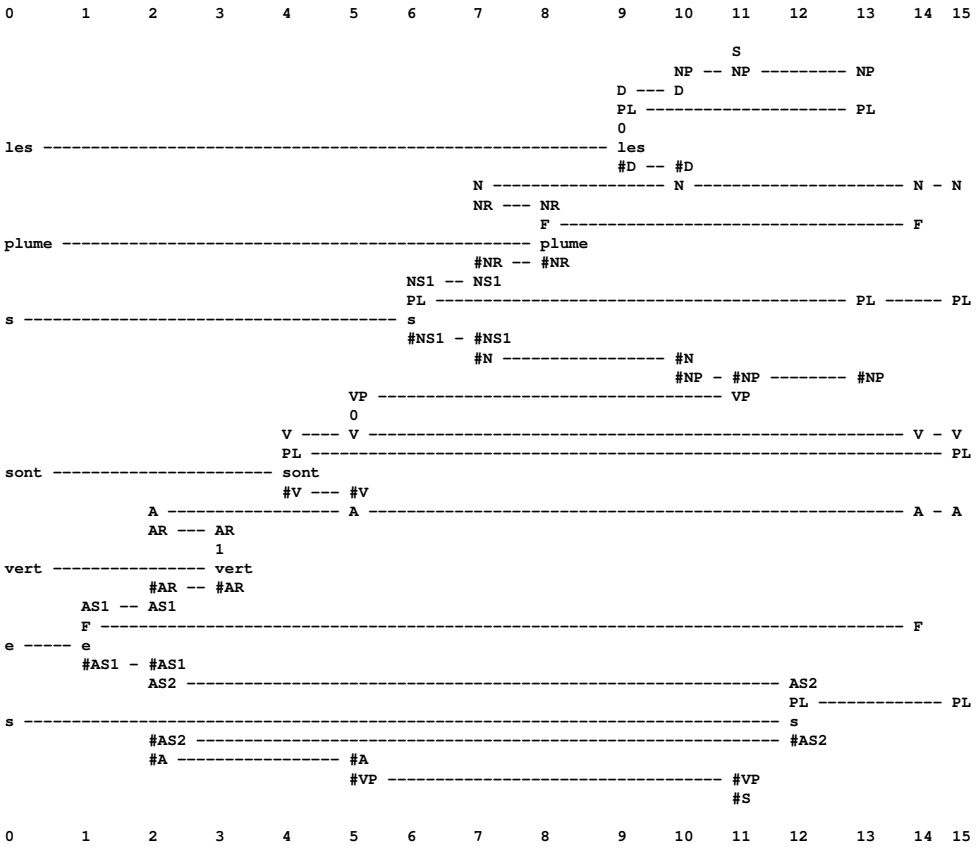

Figure 31: The best SP-Multiple-Alignment found by the SPCM with the New SP-Pattern in column 0, '`les plume s sont vert e s`', representing a sentence, and Old SP-Patterns, one in each of columns 1 to 15 representing a grammatical structure or word, drawn from the SP-Grammar shown in Figure 30. As noted in the text, number dependences and gender dependes are shown in columns 13, 14, and 15. Reproduced from [86, 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.[14] 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 SP-Patterns of constraint which interact in an interesting way.

The *primary SP-Pattern of constraint* may be expressed with this sequence of SP-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 SP-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 SP-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:

[14] This section is based on [86, Section 5.5].

- Apart from the modals, which always have the same form, the first verb in the sequence, whatever it happens to be—'H', or the first 'B', or 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 a 'M' auxiliary verb is chosen, then whatever follows it—'H', or the first 'B', or the 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').
- If a 'H' auxiliary verb is chosen, then whatever follows it—the first 'B', or the second 'B', or 'V'—must have a past tense form such as 'been', 'seen', 'gone', 'slept', 'wanted', etc. In Chomsky's *Syntactic Structures* [17], 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 the '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 as above).
- The constraints apply to questions in exactly the same way as they do to statements.

Figure 32 shows a selection of examples with the dependencies marked.

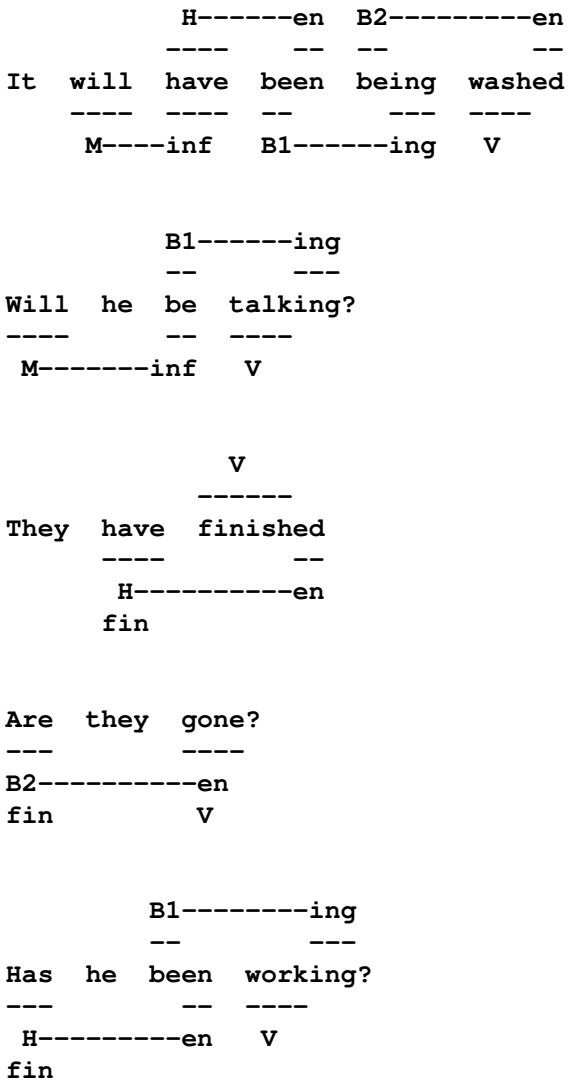

Figure 32: A selection of example sentences in English with markings of dependencies between the auxiliary verbs. *Key:* '`M`' = modal; '`H`' = forms of the verb 'have'; '`B1`' = first instance of a form of the verb 'be'; '`B2`' = second instance of a form of the verb 'be'; '`V`' = main verb; '`fin`' = 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 32 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 32) where the verb before the subject influences the form of the verb that follows immediately after the subject.

In his book *Syntactic Structures*, [17], Noam Chomsky showed that this kind of regularity in the syntax of English auxiliary verbs could be described using TG. In the SPTI, for each pair of SP-Symbols linked by a dependency (eg'`M inf`', '`H en`'; '`B1 ing`', '`B2 en`'), In TG, the two SP-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 TG 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 [27,55]. An example showing how English auxiliary verbs may be described using the definite clause grammar formalism may be found in [79, pp. 183-184]).

7.5.2 English auxiliary verbs in the SPTI

Figure 33 shows an SP-Grammar for English auxiliary verbs which exploits several of the ideas described above. Figures 34, 35 and 36 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 33. 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 33: 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 SP-Pattern for a statement (marked with the SP-Symbol '`ST`') and the second line is a sentence SP-Pattern for a question (marked with the SP-Symbol '`Q`'). Apart from these markers, the only difference between the two SP-Patterns is that, in the statement SP-Pattern, the SP-Symbols '`X1 #X1`' follow the noun phrase SP-Symbols ('`NP #NP`'), whereas in the question SP-Pattern they precede the noun phrase SP-Symbols. As can be seen in the examples in Figures 34, 35 and 36, the pair of SP-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 34 it follows the noun phrase, while in Figures 35 and 36 it precedes the noun phrase.

Each of the next four SP-Patterns in the SP-Grammar have the form '`X1 ... #X1 XR ... #S`'. The SP-Symbols '`X1`' and '`#X1`' align with the same pair of SP-Symbols in the sentence SP-Pattern. The SP-Symbols '`XR ... #S`' encode the remainder of the sequence of verbs.

The first '`X1`' SP-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`' SP-Pattern is for sentences which contain a main verb without any auxiliaries.

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

Notice that the SP-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 SP-Symbols '`XB`' and '`XV`' is left vacant. Figure 34 illustrates the case where the verb sequence starts with the first '`B`' verb with the omission of the second '`B`' verb. Figure 36 illustrates the case where the verb sequence starts with the second '`B`' verb (and the first '`B`' verb has been omitted).

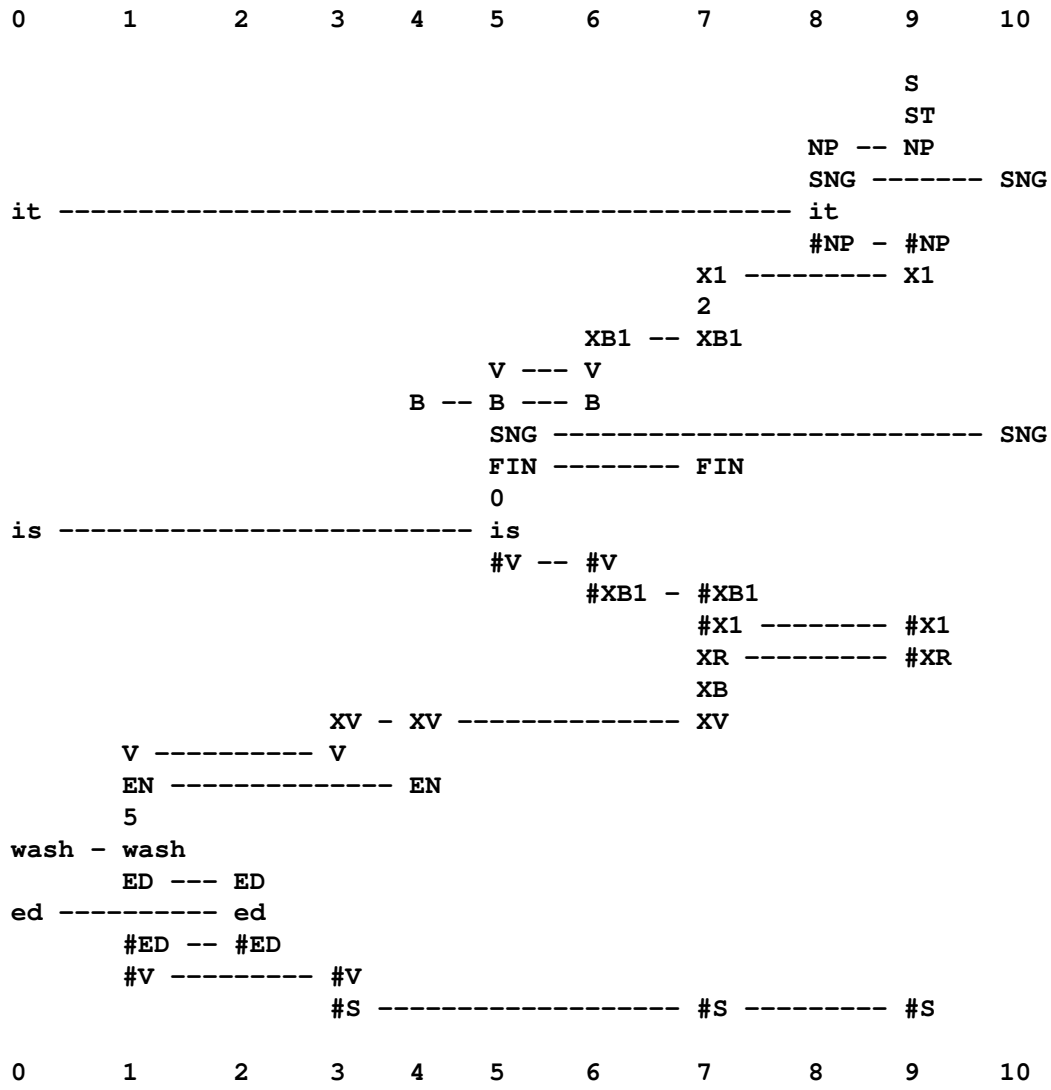

Figure 34: The best SP-Multiple-Alignment found by the SPCM with '`it is wash ed`' in New and the SP-Grammar from Figure 33 in Old. Reproduced from [86, Figure 5.12].

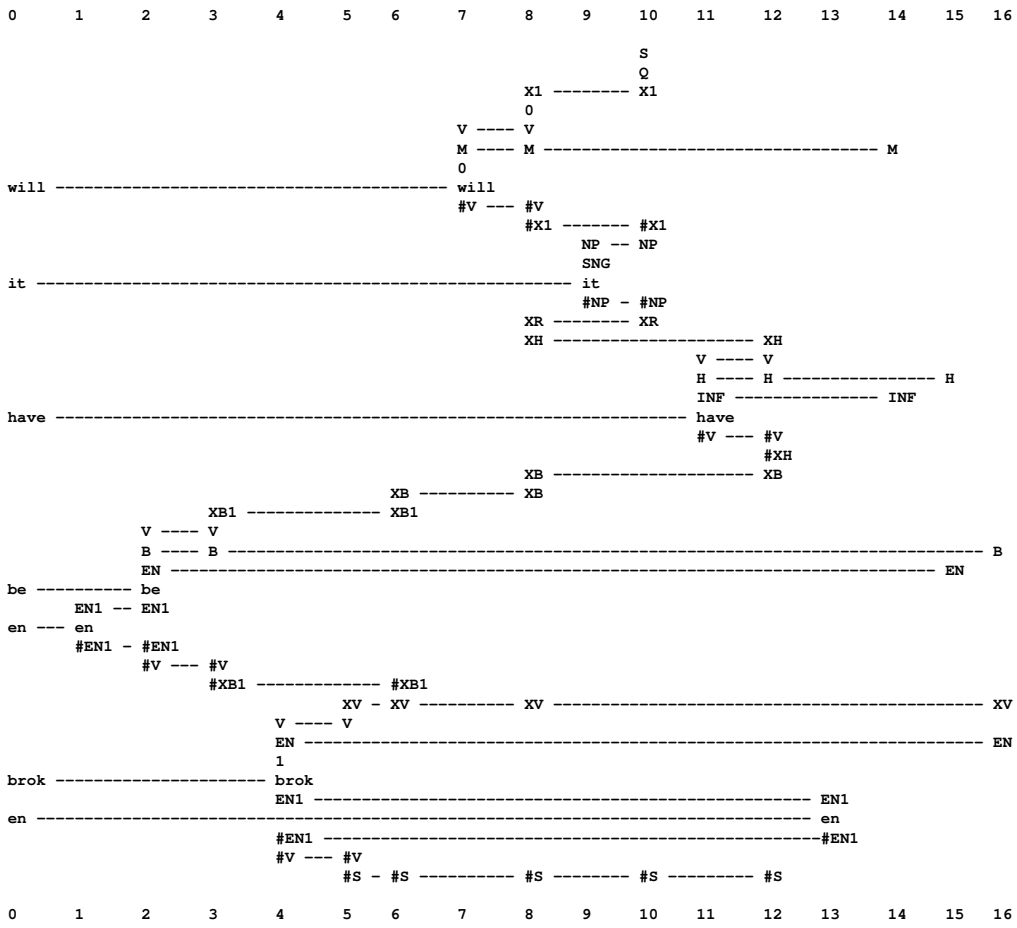

Figure 35: The best SP-Multiple-Alignment found by the SPCM with '`will it have be en brok en`' in New and the SP-Grammar from Figure 33 in Old. Reproduced from [86, Figure 5.13].

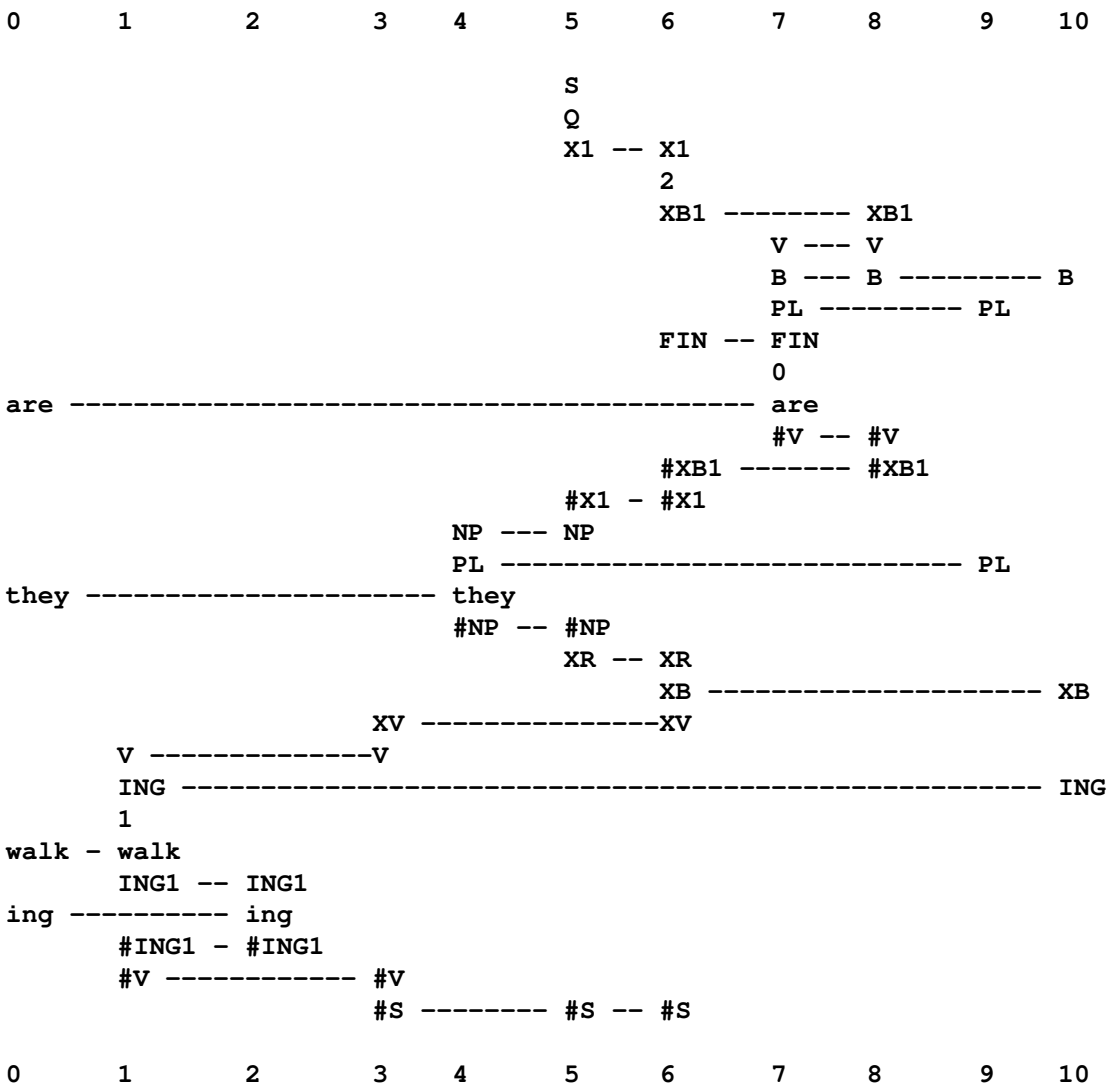

Figure 36: The best SP-Multiple-Alignment found by the SPCM with '`are they walk ing`' in New and the SP-Grammar from Figure 33 in Old. Reproduced from [86, Figure 5.14].

7.5.4 The secondary constraints

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

Examples appear in all three SP-Multiple-Alignments in Figures 34, 35 and 36. In every case except one (column 4 in Figure 34), the SP-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 34) or whether they are separated from each other by the subject noun-phrase (e.g., the question in Figure 35 and in Figure 36). Notice, again, how the overlapping dependencies in Figure 35 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 8.2, 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.[15]

A preliminary example of how this might be done is shown in Figure 37. This is the best SP-Multiple-Alignment produced by the SPCM with 'john kissed mary' as the New SP-Pattern and an SP-Grammar in Old that contains SP-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.

[15]This section is based on [86, Section 5.7].

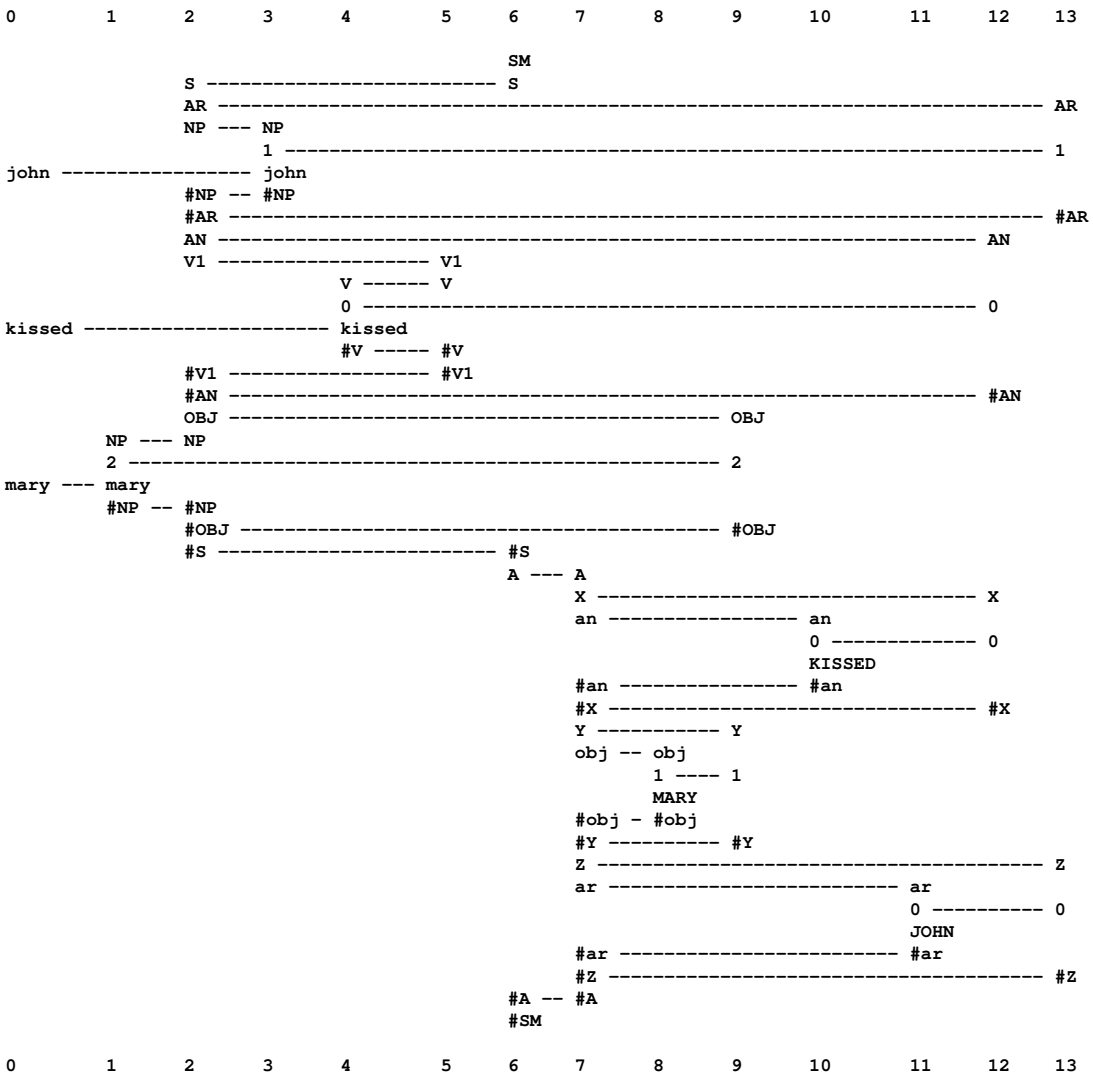

Figure 37: The best SP-Multiple-Alignment created by the SPCM with the sentence 'john kissed mary' in New and a grammar in Old that represents natural language syntax, semantics and their integration. Reproduced from [86, Figure 5.18].

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

- The SP-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 SP-Pattern differs from comparable SP-Patterns shown in previous examples because each constituent within the SP-Pattern is marked with its semantic role. Thus the first noun phrase ('NP #NP') is enclosed by the pair of SP-Symbols 'AR ... #AR' (representing the 'actor' role), the verb ('V1 #V1') is marked as an 'action' with the SP-Symbols 'AN ... #AN', and the second noun phrase is marked as an 'object' ('OBJ ... #OBJ').

- In column 7, the SP-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 SP-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.

- In column 6, the SP-Pattern 'SM S #S A #A #SM' provides a link between the SP-Pattern in column 2 representing the syntactic structure of the sentence ('S ... #S') and the action-object-actor SP-Pattern in column 7 ('A ... #A').

- In columns 9, 12 and 13 are three more SP-Patterns that link the syntax with the semantics. In column 9, the SP-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 SP-Pattern in column 12 connects 'kissed' with 'KISSED' in the 'an #an' semantic slot; and the SP-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 of the linguistic knowledge 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 8.7, 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 8.7. 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 38 shows how, via the building of an SP-Multiple-Alignment, a sentence may be derived from the SP-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 SP-Pattern in New and the same grammar in Old as was used for the example in Figure 37. 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' SP-Symbols in the SP-Multiple-Alignment, we have the sentence corresponding to the semantic representation in column 0.

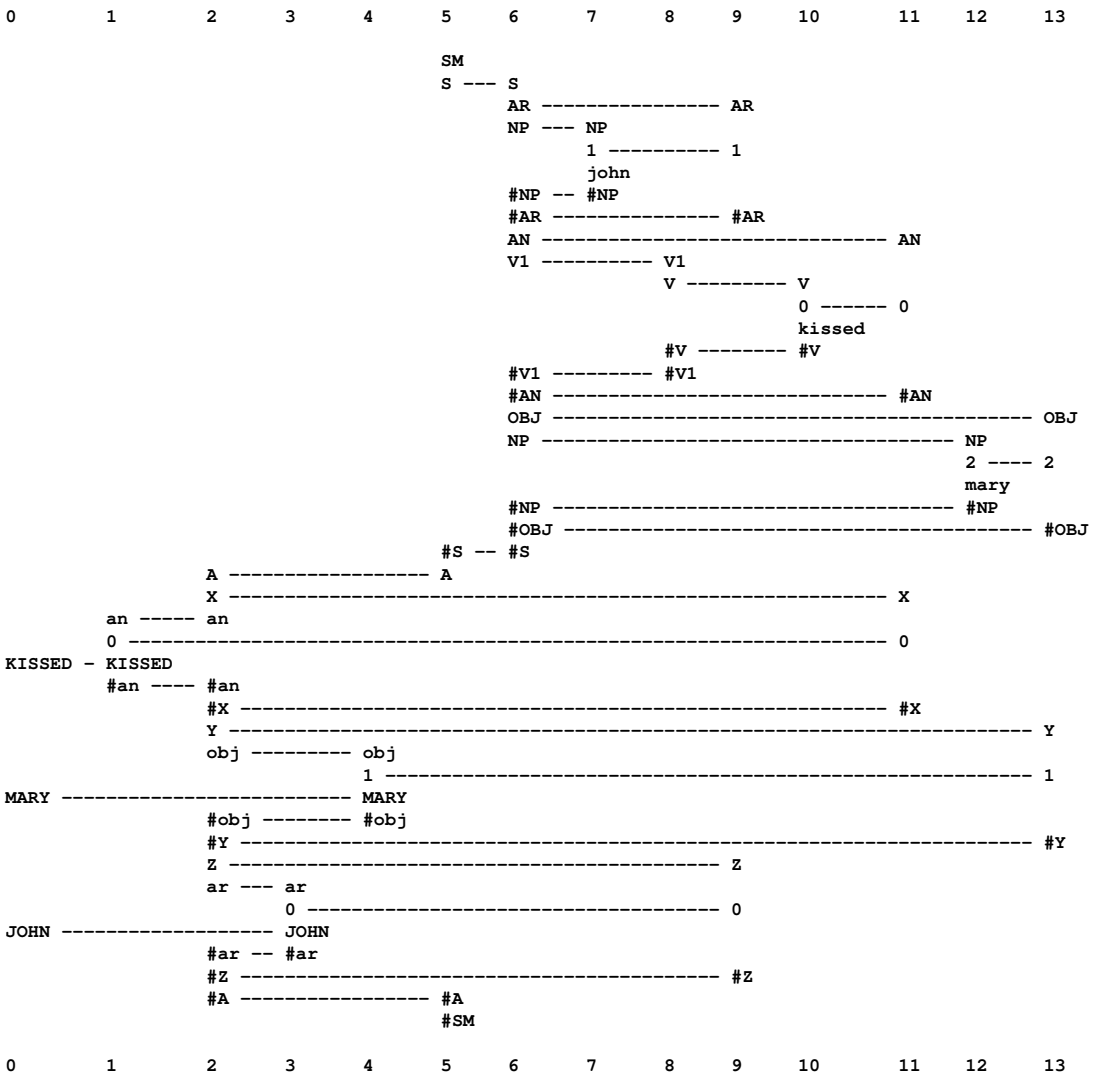

Figure 38: The best SP-Multiple-Alignment created by the SPCM with 'KISSED MARY JOHN' in New and the same grammar in Old as was used for the example shown in Figure 37. Reproduced from [86, Figure 5.19].

7.7 Recognition and retrieval

In this section, Figure 39 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.[16]

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), in the order 'Ranunculales' (column 4,) and so on.

This process of recognition may also be regarded as a process of information retrieval because it retrieves information that was not in the SP-Pattern in column 0, amongst the New 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.

[16] This section is based on [89, Section 10.1].

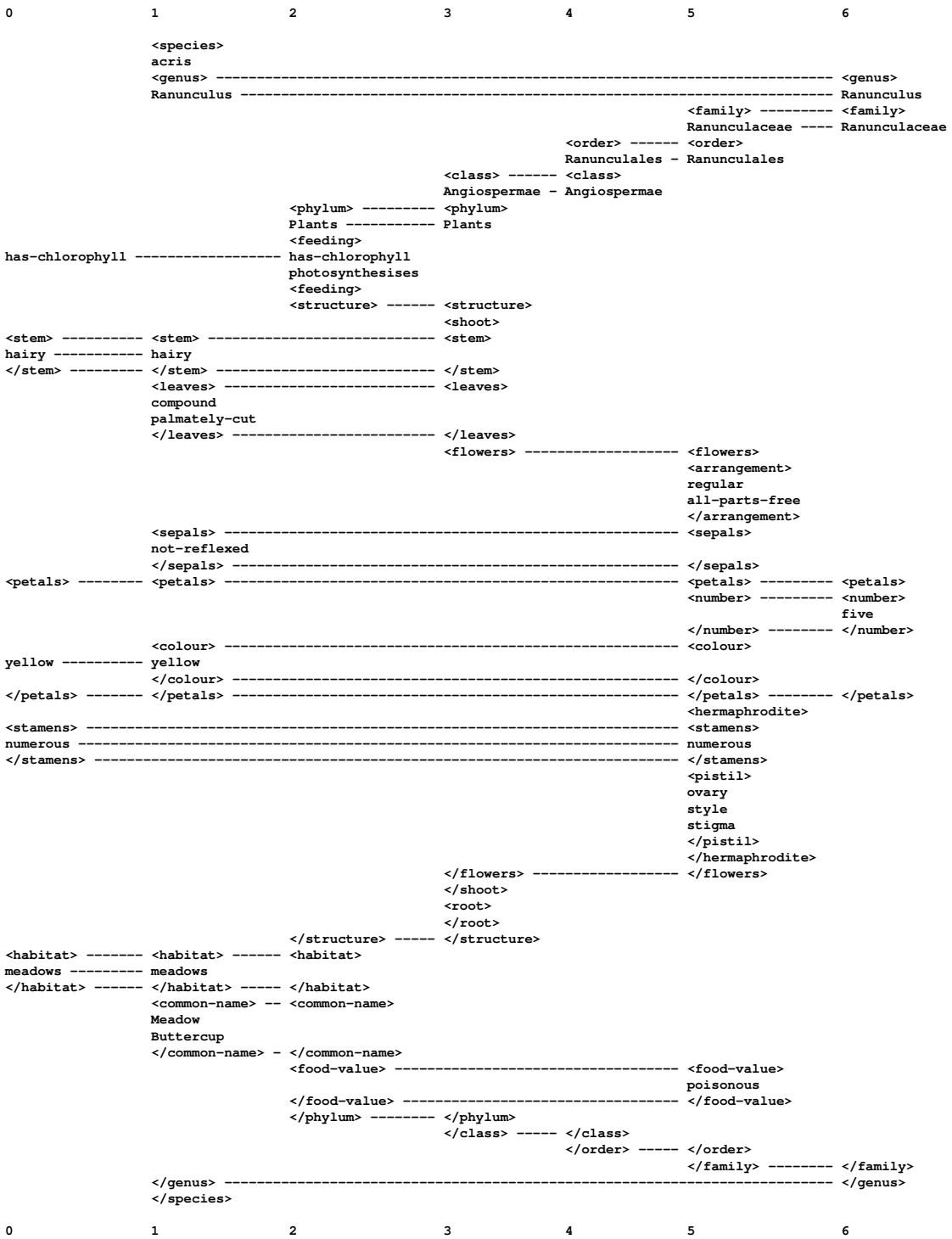

Figure 39: The best SP-Multiple-Alignment created by the SPCM, with a set of New SP-Patterns (in column 0) that describe some features of an unknown plant, and a set of Old SP-Patterns, including those shown in columns 1 to 6, that describe different categories of plant, with their parts and sub-parts, and other attributes. Reproduced from [89, 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 40: The set of New SP-Patterns supplied to the SPCM for the example discussed in the text. These SP-Patterns represent the patient 'John Smith' and his symptoms.

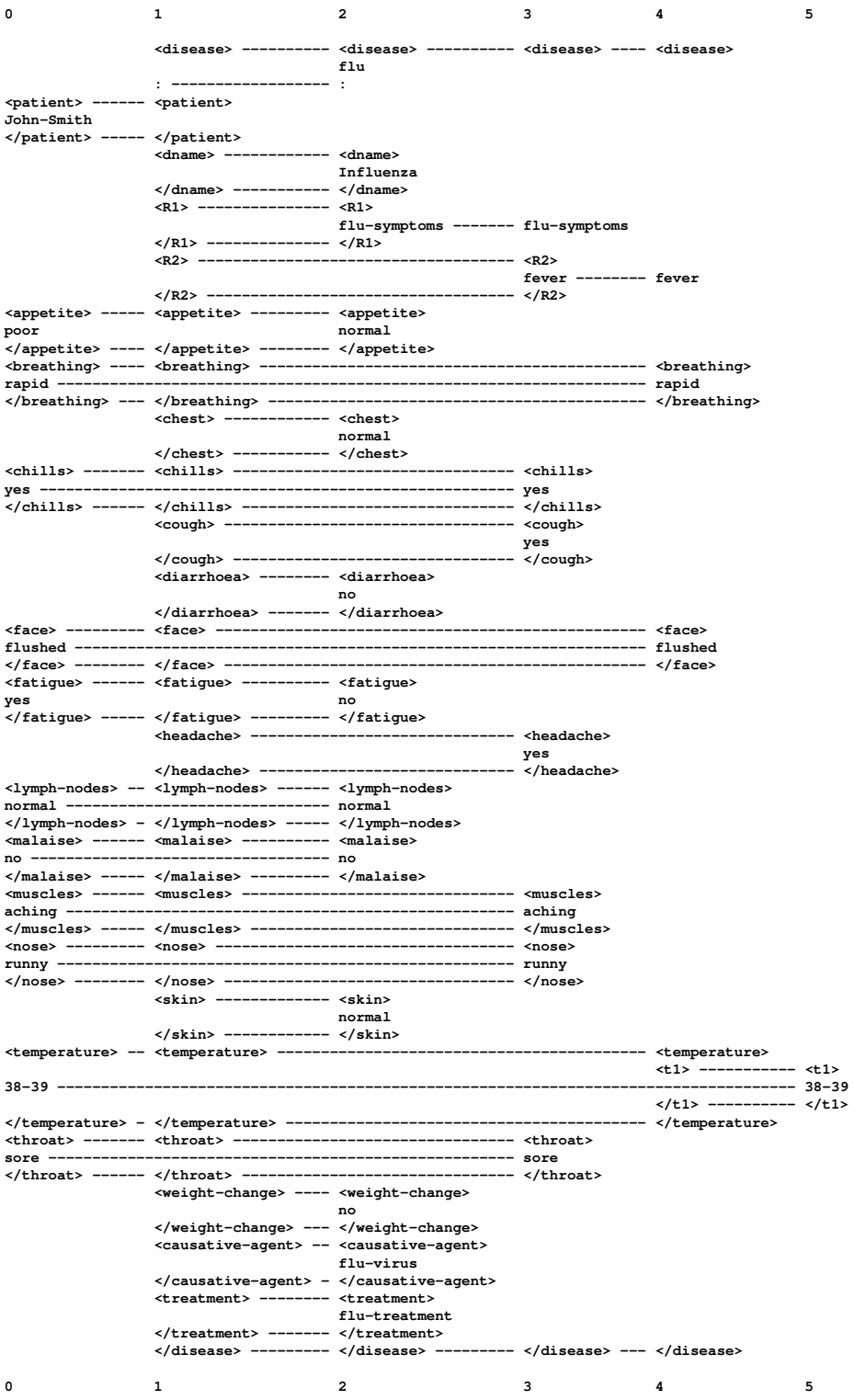

Figure 41: The best SP-Multiple-Alignment found by the SPCM with the set of SP-Patterns from Figure 40 in New (describing the symptoms of the patient 'John Smith') and a set of SP-Patterns in Old describing a range of different diseases and named clusters of symptoms, together with the 'framework' SP-Pattern shown in column 1. Reproduced from [85, Figure 6].

The likely diagnosis in this case is that the patient, John Smith, has flu (column 3).

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.[17] 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 Figures.

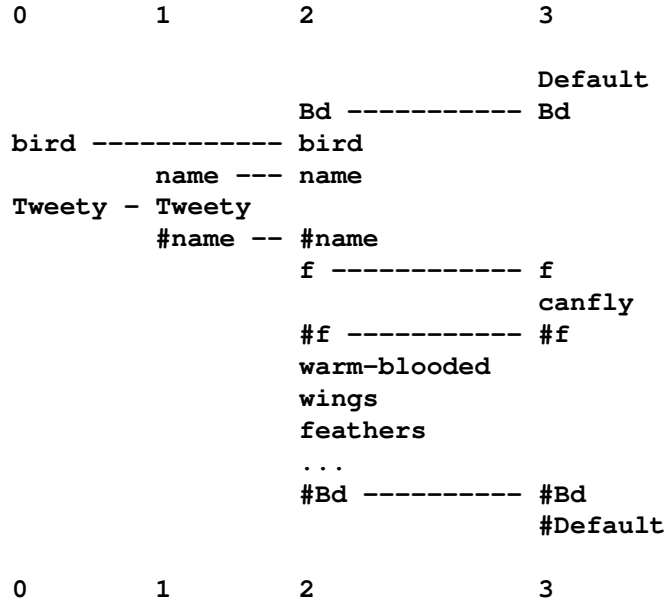

Figure 42: The first of the three best SP-Multiple-Alignments formed by the SPCM with the SP=Pattern 'bird Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is calculated as 0.66. Reproduced from [89, Figure 17].

In Figure 42, a bird called 'Tweety' (columns 0 and 1) is identified as a bird (column 2) and, as such, it can (probably) fly (column 3). This inference is prob-

[17] This section is based on [89, Section 10.1].

abilistic because, as described below, there are two other SP-Multiple-Alignments that can be formed from the information that Tweety is a bird (Figures 43 and 44). 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).

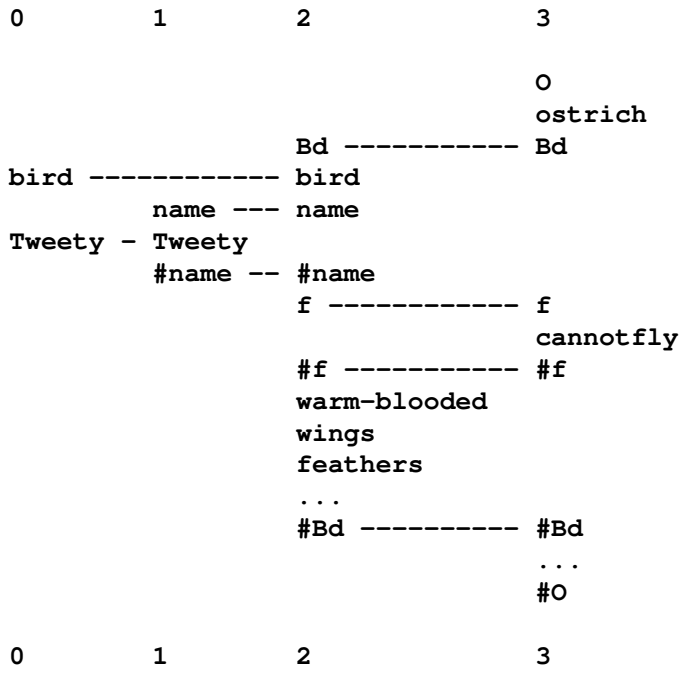

Figure 43: The second of the three best SP-Multiple-Alignments formed by the SPCM with 'bird Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is calculated as 0.22. Reproduced from [89, Figure 18].

In Figure 43, 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.

Figure 44, is much the same as Figure 43 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.

Figure 45 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

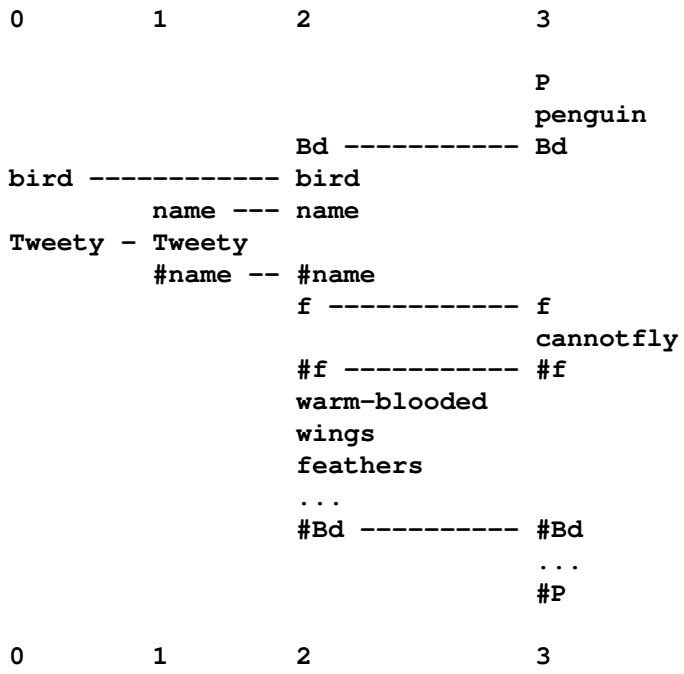

Figure 44: The last of the three best SP-Multiple-Alignments formed by the SPCM with 'bird Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 0.12.

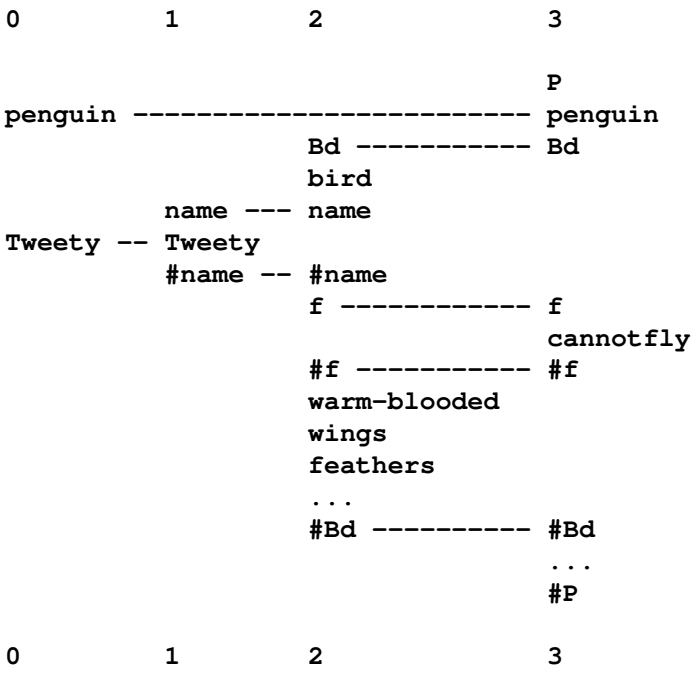

Figure 45: The best SP-Multiple-Alignment formed by the SPCM with 'penguin Tweety' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 1.0.

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 [54].

A Bayesian network is a directed, acyclic graph like the one shown in Figure 46 (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 [54, 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. [54, 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 7.12.1 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 46.

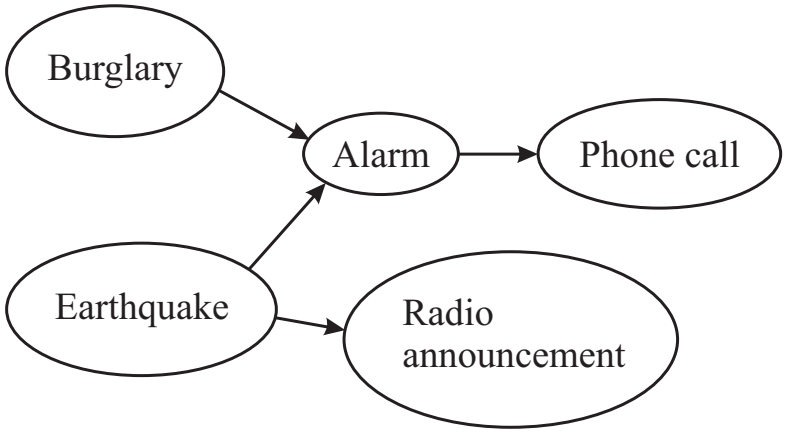

Figure 46: A Bayesian network representing causal relationships discussed in the text. In this diagram, 'Phone call' means 'a phone call about the alarm going off' and 'Radio announcement' means 'a radio announcement about an earthquake'.

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).

7.10.2 Representing contingencies with SP-Patterns and frequencies

To see how representing 'explaining away' may be understood in terms of the SPTI, consider, first, the set of SP-Patterns shown in Figure 47, 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.

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

Figure 47: A set of SP-Patterns to be stored in Old in an example of 'explaining away'. The SP-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 SP-Symbols 'e1' and 'e2' represent other contexts for 'earthquake' besides the contexts 'alarm' and 'radio-earthquake-announcement'.

The fourth SP-Pattern in Figure 47 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 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 47 are intended to reflect the two

probabilities suggested for this example in [54, 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 SP-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 47 include:

- No attempt has been made to represent the idea that 'the last false alarm you recall was triggered by an earthquake' [54, p. 9]. At some stage in the development of the SPTI, there will be a need to take account of recency.

- 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).

Symbol	*Probability*
alarm	1.0
burglary	0.3281
earthquake	0.0156

Table 1: The probabilities of unmatched SP-Symbols, calculated by the SPCM for the three SP-Multiple-Alignments shown in Figure 48.

- 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 47 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 SP-Symbol 'phone-alarm-call' in New. Figure 48 shows, at the top, the best SP-Multiple-Alignment formed by the SPCM in this case with the SP-Patterns from Figure 47 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 47.

The unmatched SP-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 6.3.12) 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 SP-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 47 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

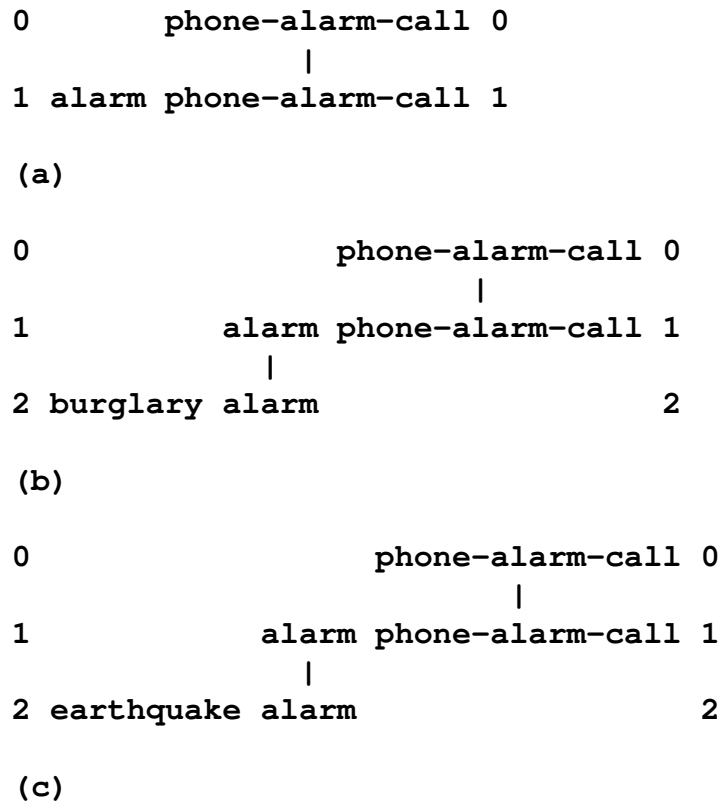

Figure 48: The best SP-Multiple-Alignment (at the top) and the other two SP-Multiple-Alignments in its reference set formed by the SPCM with the SP-Pattern 'phone-alarm-call' in New and the SP-Patterns from Figure 47 in Old. In order from the top, the values for CD with relative probabilities in brackets are: 19.91 (0.6563), 18.91 (0.3281), and 14.52 (0.0156).

where earthquakes were not vanishingly rare).

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 SP-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 49) 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 SPTI. The main point is that the SP-Multiple-Alignment(s) 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(s) 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.

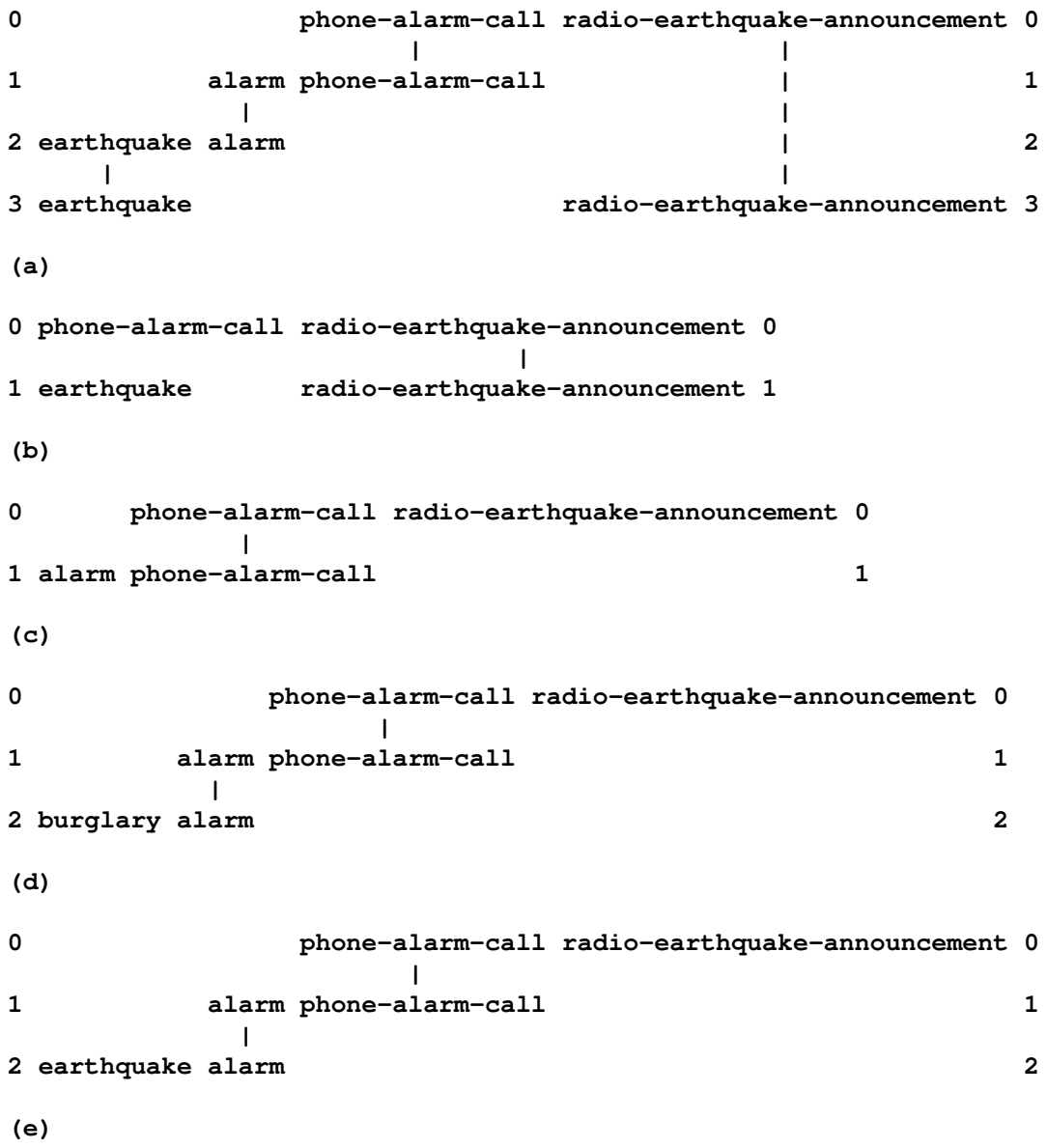

Figure 49: At the top, the best SP-Multiple-Alignment formed by the SPCM with the SP-Pattern 'phone-alarm-call radio-earthquake-announcement' in New and the SP-Patterns from Figure 47 in Old. Other SP-Multiple-Alignments formed by the SPCM are shown below. From the top, the CD values are: 74.64, 54.72, 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 SPTI 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 49 with each other, or with SP-Patterns or SP-Symbols from Old for both these things. The two examples just mentioned are shown in Figure 50.

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 50.

Given the existence of SP-Multiple-Alignments like those shown in Figure 50, values for relative probabilities of SP-Multiple-Alignments will change. The best SP-Multiple-Alignment from Figure 49 and the two SP-Multiple-Alignments from Figure 50 constitute a reference set because they all 'encode' the same SP-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 49 and the two SP-Multiple-Alignments in Figure 50, 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.

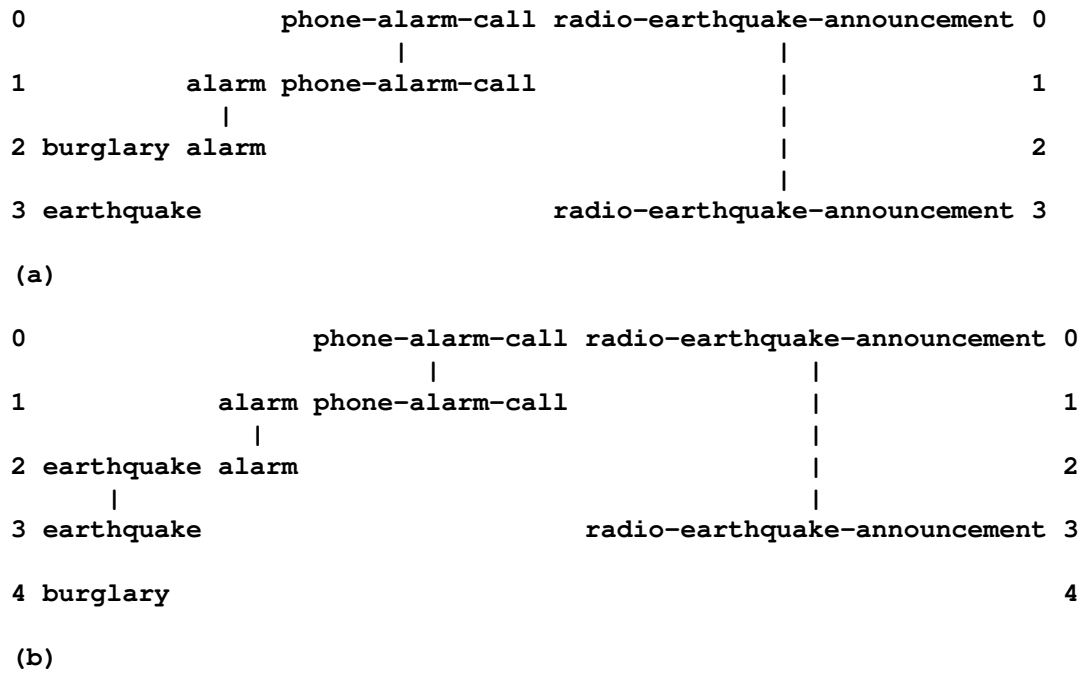

Figure 50: Two SP-Multiple-Alignments discussed in the text. (a) An SP-Multiple-Alignment created by combining the second and fourth SP-Multiple-Alignment from Figure 49. CD = 73.64, Absolute P = 5.5391e-5. (b) An SP-Multiple-Alignment created from the first SP-Multiple-Alignment in Figure 49 and the SP-Symbol 'burglary'. CD = 72.57, Absolute P = 2.6384e-5.

SP-Multiple-Alignment	*Absolute probability*	*Relative probability*
(a) in Figure 49	1.1052e-4	0.5775
(a) in Figure 50	5.5391e-5	0.2881
(b) in Figure 50	2.6384e-5	0.1372

Table 2: Values for absolute and relative probability for the best SP-Multiple-Alignment in Figure 49 and the two SP-Multiple-Alignments in Figure 50.

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 networks 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—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 6.4.1.

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 51 to 55

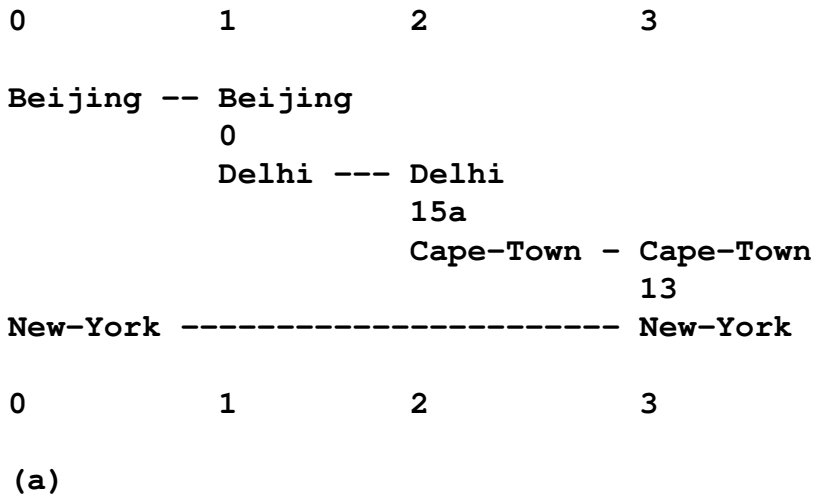

Figure 51: This and the following four figures show SP-Multiple-Alignments representing a selection of routes between Beijing and New York. They are amongst the best SP-Multiple-Alignments found by the SPCM with 'Beijing New-York' in New, and a repository of Old SP-Patterns showing one-way air links between individual cities as described in the text.

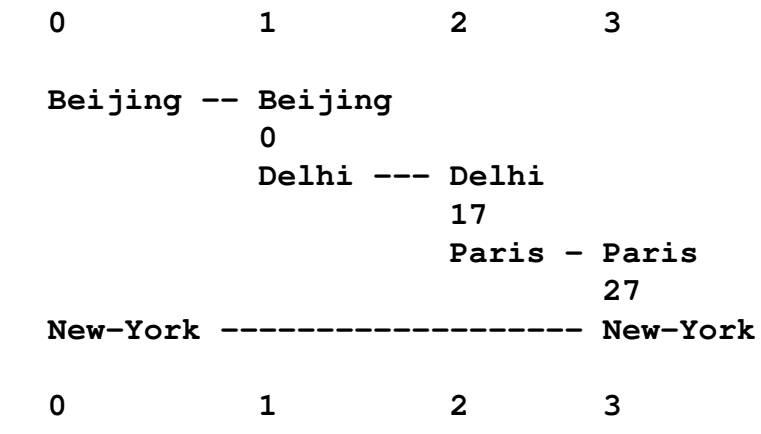

Figure 52: See Figure 51.

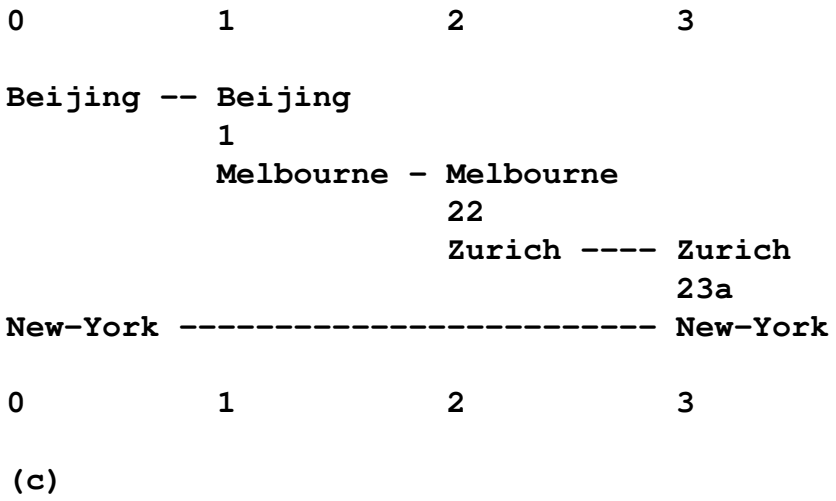

Figure 53: See Figure 51.

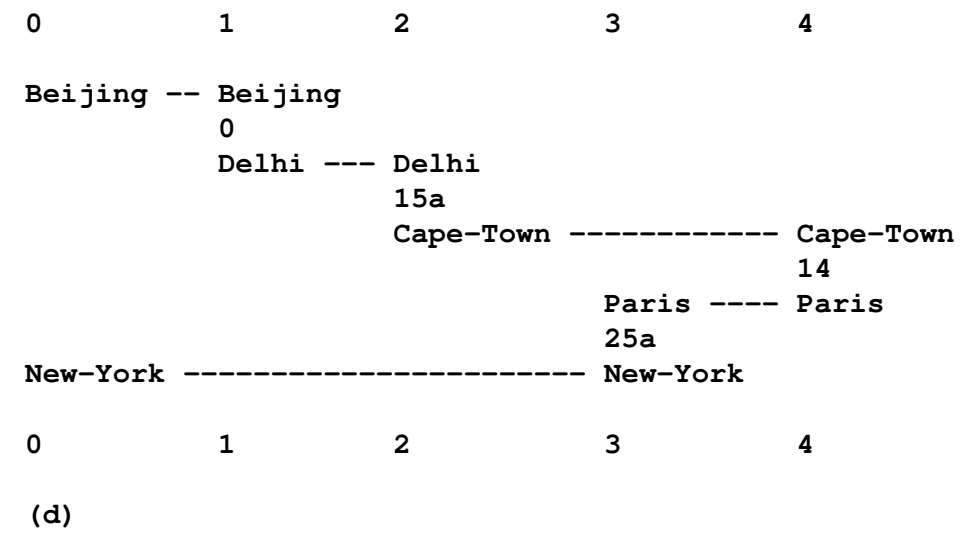

Figure 54: See Figure 51.

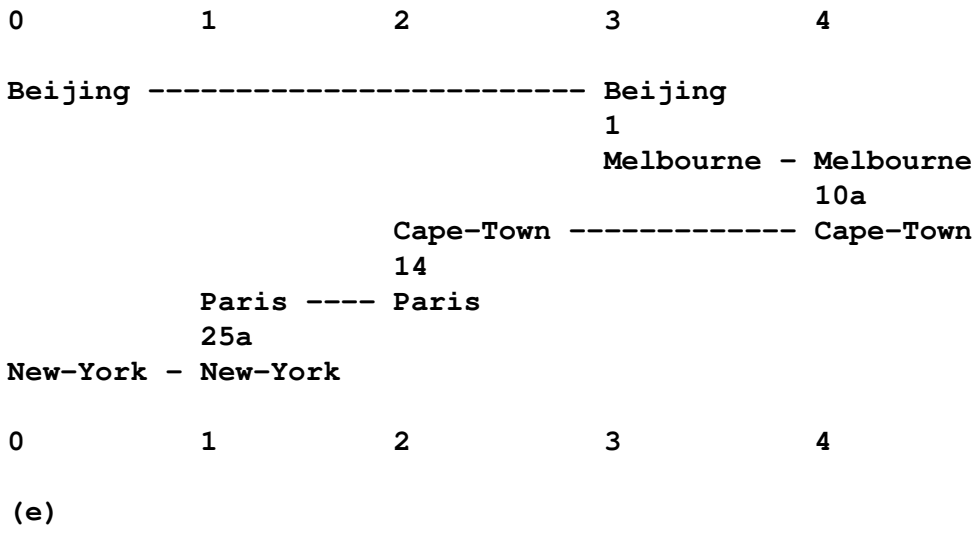

Figure 55: See Figure 51.

7.12 Problem solving

As noted in Section 4.1.3, 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, ...' [4, 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 56 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.

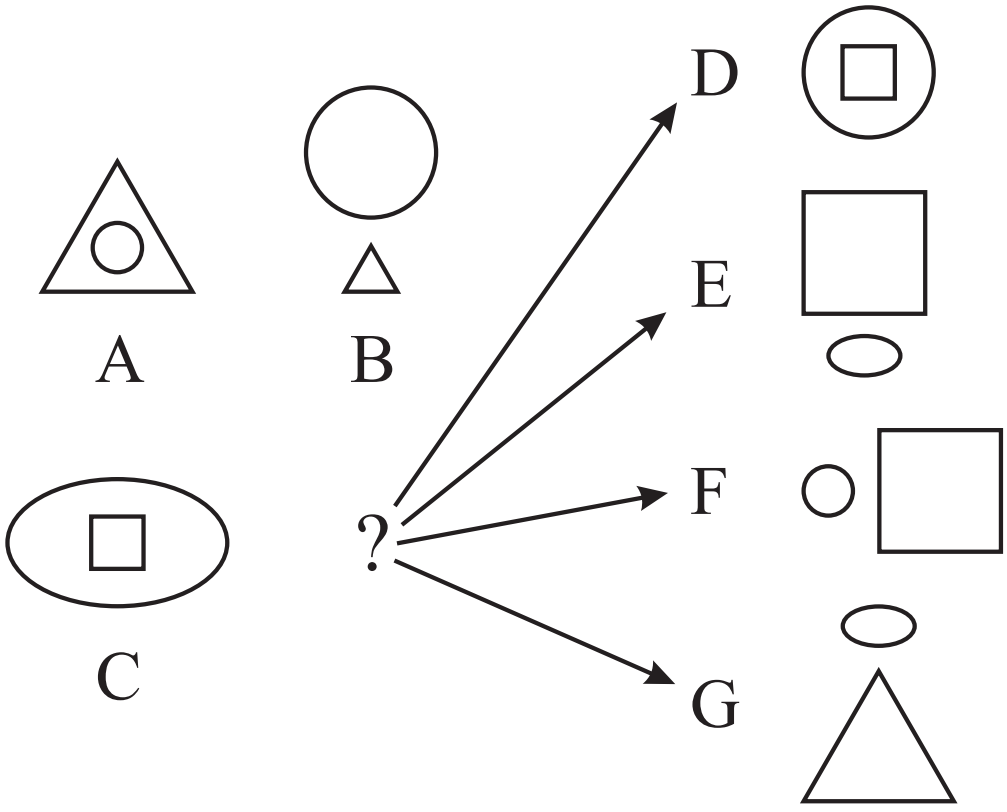

Figure 56: A geometric analogy problem.

Normally, these tests use simple geometric SP-Patterns like those shown in Figure 56 but because the SPCM has not yet been developed to process two-dimensional SP-Patterns, the geometric SP-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 [24] well-known heuristic algorithm). In more recent work [6,26], 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 SPTI.

Given that the diagrammatic form of the problem has been translated into textual SP-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 57 shows the best SP-Multiple-Alignment created by the SPCM with New information in column 0 corresponding to the geometric SP-Patterns 'A' and 'B' in Figure 56, and the Old SP-Patterns shown in Figure 58 corresponding to the geometric SP-Patterns 'D', 'E', 'F' and 'G' in Figure 56.

```
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 57: 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 Figure 57, the best SP-Multiple-Alignment found by the SPCM shows, in column 2, the combination of textual SP-Patterns corresponding to a combination of geometric SP-Patterns 'C' and 'E' in Figure 56, 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

```
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 58: Textual SP-Patterns corresponding to the combination of one of '`C1`', '`C2`', '`C3`', or '`C4`' on the bottom left of Figure 56 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.

partial matches between SP-Patterns.

7.12.1 Summary of the kinds of probabilistic reasoning exhibited by the SPTI

Several kinds of probabilistic reasoning flow from one relatively simple framework, the concept of SP-Multiple-Alignment (see [86, Chapter 7], [89, Section 10]). These include: one-step 'deductive' reasoning; abductive reasoning; reasoning with probabilistic networks and trees; reasoning with 'rules'; nonmonotonic reasoning; 'explaining away'; causal diagnosis; reasoning which is not supported by evidence; and there is potential for spatial reasoning and what-if reasoning.

To illustrate the potential of this aspect of the SPTI, here is an example showing how the SPCM may accommodate nonmonotonic reasoning.

Nonmonotonic reasoning and reasoning with default values . This section presents a simple example which shows how the SPTI can accommodate nonmonotonic reasoning (Reproduced with permission from [89, Section 10.1]).

The concepts of *monotonic* and *nonmonotonic* reasoning are well explained by [28]. In brief, conventional deductive inference 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.

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 an ostrich or a penguin (unless he or she is an astonishing new kind of ostrich or penguin that can fly).

Typically, birds fly When, for example, the SPCM is used to model one-step deductive reasoning or abductive reasoning, the idea that (all) birds can fly may be expressed with the SP-Pattern '`Bd bird name #name canfly warm-blooded wings feathers ... #Bd`'. This, of course, is an oversimplification of the real-world facts because, while it true that the majority of birds fly, we know that there are also flightless birds like ostriches, penguins and kiwis.

In order to model these facts more closely, we need to modify the SP-Pattern that describes birds to be something like this: '`Bd bird name #name f #f warm-blooded wings feathers ... #Bd`'. And, to our database of Old SP-Patterns, we need to add SP-Patterns like this:

```
Default Bd f canfly #f #Bd #Default
P penguin Bd f cannotfly #f #Bd ... #P
O ostrich Bd f cannotfly #f #Bd ... #O.
```

Now, the pair of SP-Symbols '`f #f`' in '`Bd bird name #name f #f warm-blooded wings feathers ... #Bd`' functions like a 'variable' that may take the value 'canfly' if a given class of birds can fly and 'cannotfly' when a type of bird cannot fly. The SP-Pattern '`P penguin Bd f cannotfly #f #Bd ... #P`' shows that penguins cannot fly and, likewise, the SP-Pattern '`O ostrich Bd f cannotfly #f #Bd ... #O`' shows that ostriches cannot fly. The SP-Pattern '`Default Bd f canfly #f #Bd #Default`', which has a substantially higher frequency than the other two SP-Patterns, represents the default value for the variable which is 'canfly'. Notice that all three of these SP-Patterns contains the pair of SP-Symbols '`Bd ... #Bd`' showing that the corresponding classes are all subclasses of birds.

Tweety is a bird so, probably, Tweety can fly When the SPCM is run with '`bird Tweety`' in New and the same SP-Patterns in Old as before, modified as just described, the three best SP-Multiple-Alignments found are those shown in Figures 59, 60 and 61.

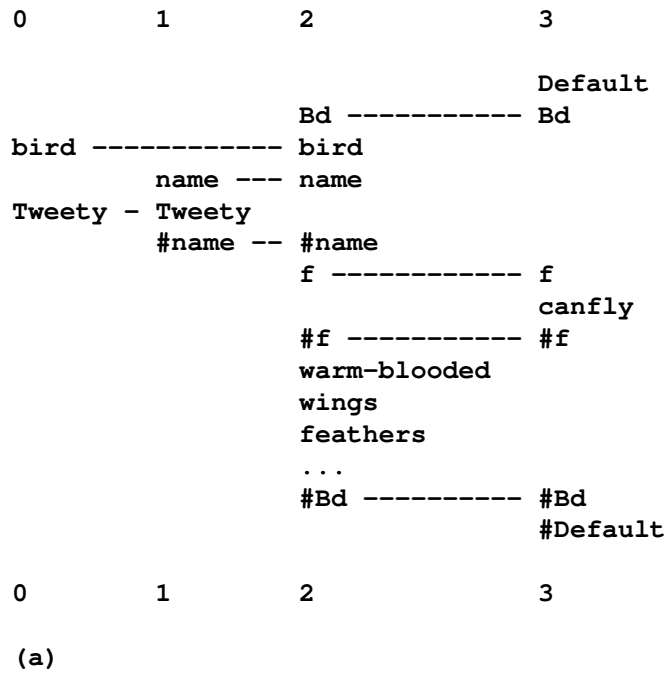

Figure 59: The first of the three best SP-Multiple-Alignments formed by the SPCM with '`bird Tweety`' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 0.66.

The first SP-Multiple-Alignment tells us that, with a relative probability of 0.66, Tweety may be the typical kind of bird that can fly. The second SP-Multiple-

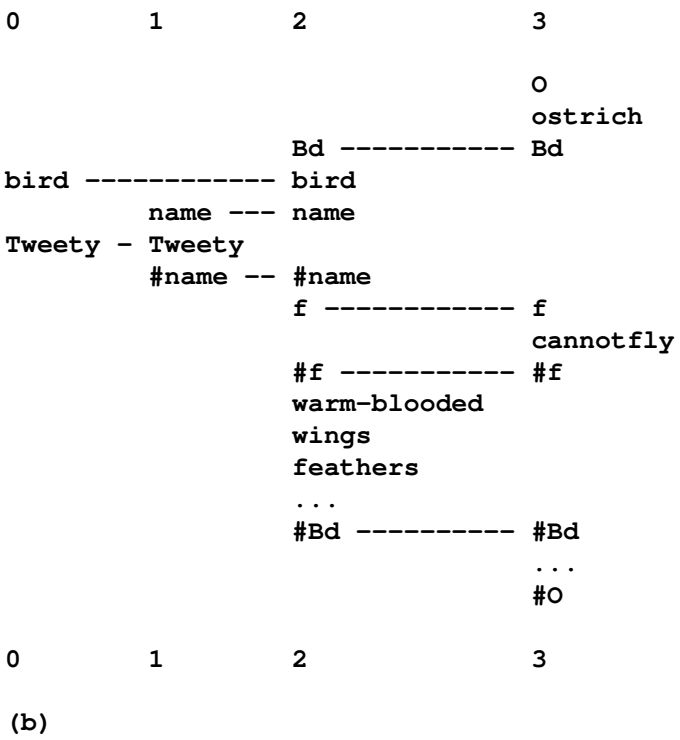

Figure 60: The second of the three best SP-Multiple-Alignments formed by the SPCM with '`bird Tweety`' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 0.22.

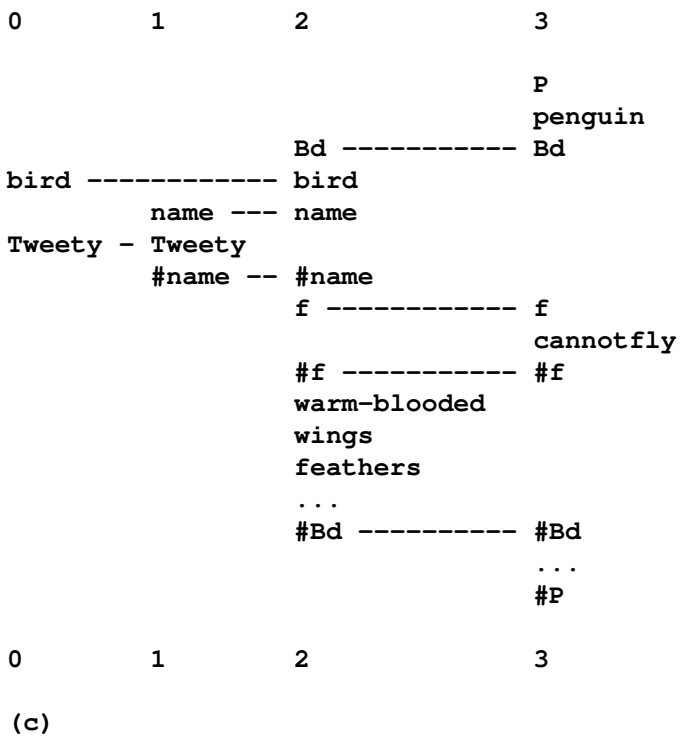

Figure 61: The last of the three best SP-Multiple-Alignments formed by the SPCM with '`bird Tweety`' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 0.12.

Alignment tells us that, with a relative probability of 0.22, Tweety might be an ostrich and, as such, he or she would not be able to fly. Likewise, the third SP-Multiple-Alignment tells us that, with a relative probability of 0.12, Tweety might be a penguin and would not be able to fly. The values for probabilities in this simple example are derived from guestimated frequencies that are, almost certainly, not ornithologically correct.

Tweety is a penguin, so Tweety cannot fly Figure 62 shows the best SP-Multiple-Alignment found by the SPCM when it is run again, with '`penguin Tweety`' in New instead of '`bird Tweety`'. This time, there is only one SP-Multiple-Alignment in the reference set and its relative probability is 1.0. Correspondingly, all inferences that we can draw from this SP-Multiple-Alignment have a probability of 1.0. In particular, we can be confident, within the limits of the available knowledge, that Tweety cannot fly.

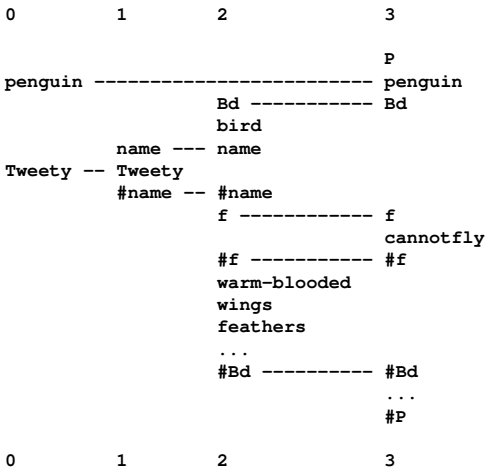

Figure 62: The best SP-Multiple-Alignment formed by the SPCM with '`penguin Tweety`' in New and SP-Patterns in Old as described in the text. The relative probability of this SP-Multiple-Alignment is 1.0.

8 Aspects of the SPTI

This Chapter describes aspects of the SPTI that don't fit easily elsewhere.

8.1 Features of the SPTI and what they are good for

Although SP-Patterns are not very expressive in themselves, they come to life in the SP-Multiple-Alignment framework within the SPTI. Within the SP-Multiple-

Alignment framework, they provide relevant knowledge for each aspect of intelligence mentioned in Section 9.1.

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 [87, Sections 3 and 4]; and relational tuples [87, Sections 3].

As noted in Section 6.2, the addition of two-dimensional SP-Patterns to the SPTI 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.

The SPTI can model concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition' ([86, Chapter 10], [93, Section 6.6.1], [96, Section 2]).

More specifically, ICMUP (Appendix G) may be seen to be the basis for much of mathematics, perhaps all of it (Chapter 10).

8.2 The seamless integration of diverse aspects of intelligence, and diverse kinds of knowledge, in any combination

An important additional feature of the SPTI, 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 kinds 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, alongside the versatility of the SPTI, seamless integration of diverse capabilities is *essential* in any artificial system that aspires to AGI:

> "AI could and should be about so much more than getting your digital assistant to book a restaurant reservation. It could and should be about curing cancer, figuring out the brain, inventing new materials that allow us to improve agriculture and transportation, and coming up with new ways to address climate change." [47, p. 21].

Figure 63 shows schematically how the SPTI, with the SP-Multiple-Alignment at centre stage, exhibits versatility in diverse aspects of intelligence and diverse

kinds of knowledge, and their seamless integration.

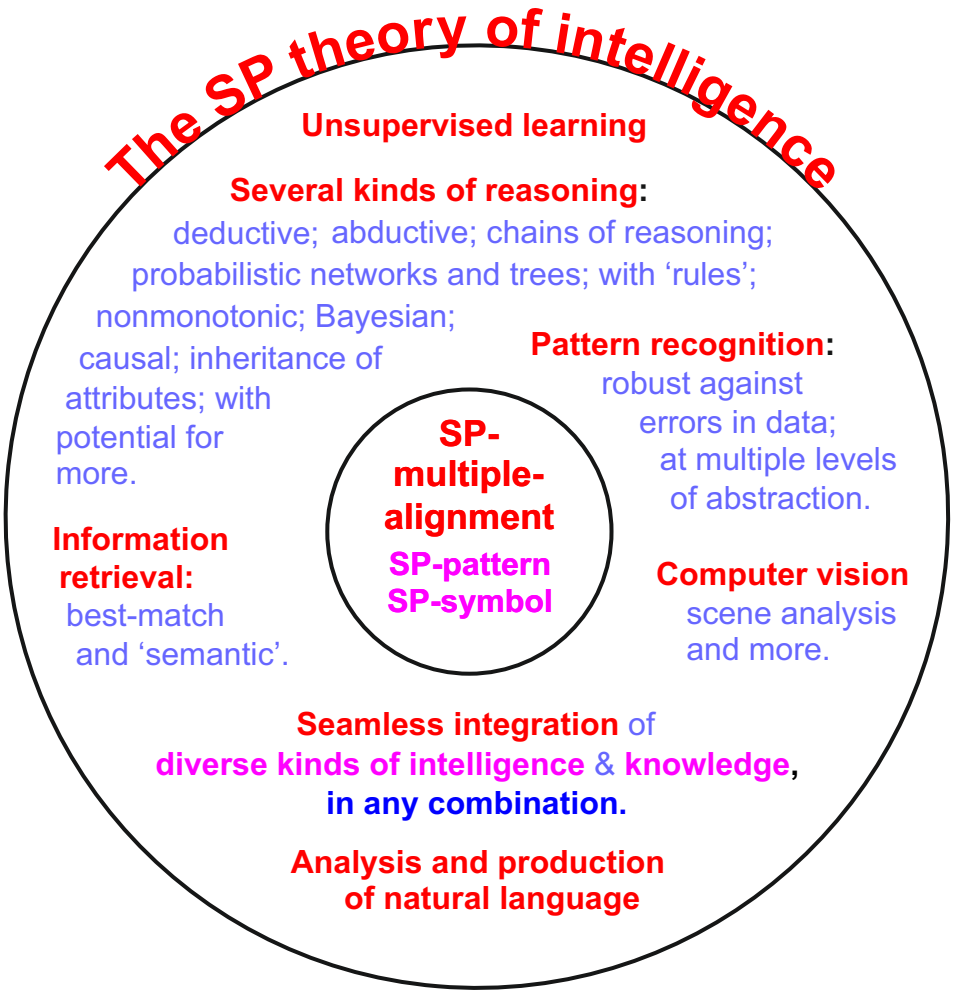

Figure 63: A schematic representation of versatility and seamless integration in the SPTI, with the SP-Multiple-Alignment concept, and the SP-Pattern and SP-Symbol concepts, centre stage.

8.3 The clear potential of the SPTI to solve 23 significant problems in AI research

Strong evidence in support of the SPTI has arisen, indirectly, from the book *Architects of Intelligence* by science writer Martin Ford [25]. 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:

> "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." [25, p. 2].

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 SPTI perspective because, with 17 of those problems, with 3 others not described in the book, 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 following summary describes the 20 problems briefly, with 3 others, each problem with a summary of how the SPTI may solve it.

Items that describe advantages of the SPTI compared with DNNs, 21 in all, are marked with '`[DNN]`'.

1. *The SP-Symbolic versus sub-SP-Symbolic divide* [DNN]. The need to bridge the divide between symbolic and sub-symbolic kinds of knowledge and processing [105, Section 2]. The concept of an SP-Symbol (see Section 6.2) 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* [DNN]. The tendency of DNNs to make large and unexpected errors in recognition [105, Section 3]. The overall workings of the SPTI and its ability, in the recognition of patterns, to correct errors in data (Section 7.3) suggests that it is unlikely to suffer from these kinds of error.

3. *Natural languages* [DNN]. 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 [105, Section 4]. The SPTI has clear potential in the representation and processing of several aspects of natural language (Section 9.1.4).

4. *Unsupervised learning* [DNN]. 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 [105, Section 5]. Learning in the SPTI is entirely unsupervised (Section 6.4).

It is clear that most human learning, including the learning of our first language or languages [80], is achieved via unsupervised learning, without the kinds of assistance described in Section 6.4.

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 reinforcement learning, learning by imitation, learning by being told, and so on (Section 6.4).

5. *Generalisation, over-generalisation, and under-generalisation* [DNN]. The need for a coherent account of generalisation, over-generalisation (under-fitting) and under-generalisation (over-fitting) (Section 6.4.5, [105, Section 6]).

6. *Reduce or eliminate the corrupting effect of 'dirty data' in unsupervised learning* [DNN]. Although this is not mentioned in Ford's book [25], there is the problem of reducing or eliminating the corrupting effect of errors in the data which is the basis for unsupervised learning.

7. *One-Shot Learning* [DNN]. 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 [105, Section 7].

8. *Transfer learning* [DNN]. Although transfer learning—incorporating old learning in newer learning—can be done to some extent with DNNs [62, Section 2.1], DNNs fail to capture the fundamental importance of transfer learning for people, or transfer learning's integral and important part of how the SPTI works (Section 6.4.1, [105, Section 8]).

9. *Reduced demands for data and for computational resources compared with DNNs* [DNN]. The ability of the SPTI to learn from a single exposure or experience (above), and the fact that transfer learning is an integral part of how it works (above), is likely to mean that, compared with DNNs, the SPTI will make greatly reduced computational demands and greatly reduced demands for data [105, Section 9].

10. *Transparency* [DNN]. By contrast with DNNs, which are opaque both in how they represent knowledge and how they process it, the SPTI is entirely transparent in both the representation and processing of knowledge [105, Section 10].

11. *Probabilistic reasoning* [DNN]. The SPTI is entirely probabilistic in all its inferences, including the forms of probabilistic reasoning described in [86, Chapter 7] and [89, Section 10].

12. *Commonsense reasoning and commonsense knowledge* [DNN]. Unlike probabilistic reasoning, the area of commonsense reasoning and commonsense knowledge is surprisingly challenging [22]. With qualifications, the SPTI shows some promise in this area [84, 97], [105, Section 12].

13. *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. The SPTI has potential in this area ([103], [105, Section 13]).

14. *Compositionality in the representation of knowledge* [DNN]. DNNs are not well suited to the representation of Part-Whole Hierarchies or Class-Inclusion Hierarchies. By contrast, the SPTI has robust capabilities in this area [105, Section 14].

15. *Establishing the importance of IC in AI research* [DNN]. There is a need to raise awareness of the significance of IC in AI. The importance of IC in the workings of brains and nervous systems is described in [99] and its importance in the SPTI is described in Section 6.1 in this book (see also [105, Section 15]).

16. *Establishing the importance of IC across diverse aspects of AI and human cognition* [DNN]. A point which deserves emphasis which was not mentioned in [105] is that, while there is some recognition amongst other researchers of the importance of IC in machine learning, there appears to be less recognition of the importance of IC in other aspects of intelligence (see also Section 1.3).

    The importance of IC in the SPTI across several aspects of intelligence is a major strength of the SPTI.

17. *Establishing the importance of a biological perspective in AI research* [DNN]. There is a need to raise awareness of the importance of a biological perspective in AI research. This is very much part of the SPTI research [105, Section 16]. Although DNNs are founded on artificial neural networks, it is generally recognised that ANNs are only loosely related to real neural networks.

18. *Distributed versus localist representations for knowledge* [DNN]. A persistent issue in studies of human learning, perception, 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. DNNs employ a

distributed form for knowledge, but the SPTI, which is firmly in the localist camp, has distinct advantages compared with DNNs. This is in keeping with other evidence for localist representations in brains [105, Section 17].

19. *The learning of structures from raw data* [DNN]. DNNs are weak in the learning of structures from raw data, either linuistic or non-linguistic. By contrast, this is a clear advantage in the workings of the SPTI [105, Section 18], [59].

20. *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. In the quest for AGI, there are clear advantages in the top-down strategy which has been adopted in the SP research ([105, Section 19], [84]).

21. *Overcoming the limited scope for adaptation in deep neural networks* [DNN]. 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. By contrast, the SPTI, like people, can learn multiple concepts at any one time, 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 [105, Section 20].

22. *The problem of catastrophic forgetting* [DNN]. 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. There is no such problem with the SPTI which may store new learning independently of old learning, or form composite structures which preserve both old and new learning, in the manner of transfer learning (above) [105, Section 21].

23. *A weakness of DNNs not mentioned in [105]* [DNN]. A matter which has become increasingly clear with further thought is that, despite the impressive things that have been done with DNNs,[18] DNNs are relatively restricted in the aspects of intelligence that they can model, without augmentation. They show little of the versatility of the SPTI in modelling diverse aspects of intelligence (Chapter 9).

[18]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.

8.4 The SPTI as a foundation for the development of artificial general intelligence

More evidence in support of the SPTI is presented in the peer-reviewed paper [84]. 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 'DALL E 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, perception, and cognition [99].

8.5 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 8.6 and described quite fully by Ernest Davis and Gary Marcus in [22].

Preliminary, unpublished papers about how the SPTI may be applied in this area of research may be downloaded via links from [97, 98].

8.6 Papers about potential benefits and applications of the SPTI

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

1. *The SP Theory of Intelligence: benefits and applications*, [93]. The SPTI promises deeper insights and better solutions in several areas of application including: unsupervised learning, natural language processing, autonomous robots, 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.

2. *Autonomous robots and the SP Theory of Intelligence*, [91]. The SPTI opens up a radically new approach to the development of intelligence in autonomous robots.

3. *Big data and the SP Theory of Intelligence*, [92]. 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 SP-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.

4. *Commonsense reasoning and commonsense knowledge*, Section 8.5. Largely because of research by Ernest Davis and Gary Marcus (see, for example, [22]), the challenges in this area of AI research are now better known. Preliminary work shows that the SPTI has promise in this area.

5. *Towards an intelligent database system founded on the SP Theory of Computing and Cognition*, [87]. The SPTI has potential in the development of an intelligent database system with several advantages compared with traditional database systems.

6. *Medical diagnosis as pattern recognition in a framework of information compression by multiple alignment, unification and search*, [85]. 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.

7. *Natural Language Processing*, [86, Chapter 5], [89, Section 8]). The SPTI has strengths in the processing of natural language.

8. *How the SP System may promote sustainability in energy consumption in IT systems*, [101]. The SP Machine (Section 6.7), 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.

9. *Transparency and granularity in the SP Theory of Intelligence and its realisation in the SP Computer Model*, [104]. The SPTI with the SPCM provides an audit-trail for all its processing, and complete transparency in the way its output is structured.

10. *Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision*, [90,102]. 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.

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

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

For example, with natural language, processes for the analysis of a sentence (Section 6.3.2) are, without any qualification, the same as may be used for the production of the same sentence (Section 6.3.10).

Since the SPTI works by compressing information, this feature of the SPTI 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 [86, Section 3.8] and [89, Section 4.5] (see also Section 6.3.10).

9 Summary of the strengths of the SPTI and its realisation in the SPCM

In this chapter, a 'strength' of the SPTI is mainly a strength that has been demonstrated with the SPCM but it also means a strength that is clear from the organisation and workings of the SPTI.

9.1 Summary of AI-related strengths of the SPTI

The AI-related strengths and potential of the SPTI are summarised in the subsections that follow. Further information may be found in [89, Sections 5 to 12], [86, Chapters 5 to 9], [95], and in other sources referenced in the subsections that follow.

As we have seen in Sectiom 6.4, the SPTI has strengths in the 'unsupervised' learning of new knowledge.

The SPTI also has strengths in other aspects of intelligence including: the analysis and production of natural language; pattern recognition that is robust in the face of errors in data; pattern recognition at multiple levels of abstraction; computer vision [90]; best-match and semantic kinds of information retrieval; several kinds of reasoning (next subsection); planning; and problem solving.

9.1.1 The SPTI as a relatively firm foundation for the development of human-level AI

Notwithstanding impressive results obtained with the currently-popular DNNs, the SPTI provides a much firmer foundation for the development of AGI ([84, 105]).

9.1.2 Versatility in reasoning

.

An aspect of intelligence where the SPTI has strengths is probabilistic reasoning in several varieties including: one-step 'deductive' reasoning; chains of reasoning; abductive reasoning; reasoning with probabilistic networks and trees; reasoning with 'rules'; nonmonotonic reasoning and reasoning with default values; Bayesian reasoning with 'explaining away'; causal reasoning; reasoning that is not supported by evidence; the inheritance of attributes in class hierarchies; and inheritance of contexts in part-whole hierarchies.

There is also potential in the system for spatial reasoning [91, Section IV-F.1], and for what-if reasoning [91, Section IV-F.2].

9.1.3 Versatility in the representation of knowledge

Within the framework of SPMA, SP-patterns may serve in the representation of several different kinds of knowledge, including: the syntax of natural languages; class-inclusion hierarchies; part-whole hierarchies; discrimination networks and trees; if-then rules; entity-relationship structures; relational tuples, and concepts in mathematics, logic, and computing, such as 'function', 'variable', 'value', 'set', and 'type definition'.

There will be more potential when the SPCM has been generalised for two-dimensional SP patterns.

9.1.4 Summary of natural language processing via the SPTI

([86, Chapter 5], [89, Section 8]). As readers may have been seen already, there is a fairly detailed example in Section 6.3, Figure 12, showing how the parsing of a sentence may be represented and processed with the SP-Multiple-Alignment concept.

SPTI strengths in the learning and use of natural languages 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, and the processing of that structured knowledge.

- *The integration of syntactic and semantic knowledge.* The ability to integrate syntactic and semantic knowledge, and to process that integrated knowledge. There are simple examples in Section 7.6 and [86, Section 5.7]).

- *Discontinuous dependencies in syntax.* The ability of the SPTI to encode discontinuous dependencies in syntax such as the number dependency (singular or plural) between the subject of a sentence and its main verb (Section 6.3.4, [86, Section 9.5.2], [89, Section 8.1]).

- *Different kinds of discontinuous dependency (e.g., number dependency and gender dependency) can co-exist without interfering with each other* ([86, Section 5.4], [89, Section 8.2]).

- *Discontinuous dependencies provide an effective means of encoding the intricate structure of English auxiliary verbs.* ([86, Section 5.5], [89, Section 8.2]).

- *Representation and processing of recursive structures in natural language.* The ability to accommodate recursive structures in natural language (see, for example [86, Figures 4.4, 4.6, 5.5, and 5.6]).

- *The production of natural language.* A point of interest here is that the SPTI 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 SPTI uses exactly the same processes for both the compression and decompression of information (Section 8.7).

9.1.5 The clear potential of the SPTI to solve 20 significant problems in AI research

The SPTI has clear potential to solve 20 significant problems in AI research [105]. The first 17 of those 20 problems have been described by influential experts in AI in interviews with science writer Martin Ford, and reported in Ford's book *Architects of Intelligence* [25]. In Section 8.3, The 20 problems are described briefly, each problem with a summary of how the SPTI may solve it.

9.2 Summary of strengths of the SPTI less closely-related to AI

Apart from the AI-related strengths of the SPTI (Section 9.1), there are other potential benefits and applications such as assisting with the management of big data Section [92], assisting with medical diagnosis [85], and others detailed on tinyurl.com/5fxhybnx.

9.2.1 IC expressed via SP-Multiple-Alignments as the basis for both the compression and decompression of knowledge

Paradoxical as it may sound, IC is the basis within the SP-Multiple-Alignment concept for both the compression and decompression of knowledge. The resolution of this apparent paradox is described in Section 6.3.10.

9.2.2 Seamless integration of diverse bodies of knowledge and diverse functions, in any combination

The SP-Multiple-Alignment concept is also the basis of the SPTI's capability for the seamless integration of diverse bodies of knowledge and diverse functions, in any combination (Section 6.3). That capability is probably essential in any system that aspires to AGI, but also has potential value for non-AI kinds of processing.

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, perception, and cognition (Chapter 4), 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 [100].

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 [100, 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 and in other areas of application (Section 10.7).

An important part of the arguments for the importance of IC in mathematics are the versions of ICMUP outlined in the following subsections.

10.1 Chunking-with-Codes in mathematics

The Chunking-with-Codes technique for IC (Appendix G.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 SP-Symbol √ 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' 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 indicates 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 digits 0 to 9, but it also applies to the number of 10s, not digits.

- 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 Appendix G.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 G.2.4, run-length-coding is where some entity, SP-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—another example of run-length coding.

10.4 Combinations of these techniques within mathematics

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 Pythagoras's theorem. It exhibits IC via run-length coding in addition and in 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 many possible values for mass and corresponding values for energy. It exhibits IC via run-length coding in multiplication and in the power notation.

Other examples of ICMUP in mathematics are described in [100, 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 [100, Section 7].

As noted in Appendix B.4, for the sake of brevity, mathematics and logic will both be referred to as 'mathematics'. Computing is discussed in Section 11.2.

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 Chapter 10), and because of the close relation between IC and concepts of probability, described in Solomonoff's APT (Section 11.1), 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:

> "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." [13, p. 80].

As indicated in this quotation, randomness in number theory is closely related to Gödel's incompleteness theorems. These are themselves closely related to the phenomenon of recursion, a feature of many formal systems (including the 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* [34].

Since it is likely that logic and computing may also be understood in terms of IC [100, 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 Chapter 10, this section provides a brief discussion, picking up on the description in [100, 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:

> "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." [76, p. 14].

and similar things have been said by others.

However, Marcus Chown, quoting Stephen Wolfram, the creator of the symbolic computer language *Mathematica*, says that:

> "... 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." [19, p. 265].

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 10.3).

- 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 [107, pp. 18–19]) as an explanation for the unreasonable effectiveness of mathematics in science. But arguably symmetry is relatively complex (Appendix G.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 Chapter 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 [100, Section 9.2.1], there are potential benefits for students, teachers, practitioners, and researchers in mathematics and science.

In brief, the potential benefits of the NM include:

- *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.
- *Creating scientific theories from data.* The automatic or semi-automatic creation of scientific theories from data [93, Section 6.10.7].
- *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 (Chapter 9), including strengths in several kinds of probabilistic reasoning (Section 7.12.1).
- *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 or AI 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 [100, 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 6.4.7), 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.

11.1 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.

11.1.1 Inference and probabilities via generalisation

The kind of 'inductive' inference called 'generalisation' is part of unsupervised learning in the SPTI (Section 6.4.5.

11.1.2 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:

> "... 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." [86, Section 7.2].

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 6.3.12) 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 (Section 7.12.1), and for their seamless integration in any combination (Section 8.2). Collectively, these several kinds of probabilistic reasoning, working together, have potential as a powerful aid to statistical inference.

11.1.3 Exploiting the asymmetry between information compression and concepts of probability

Solomonoff writes:

> "Both Huffman coding [36] and the 'information packing problem'[19], [67, p. 75] used probabilities to compress information. Algorithmic probability inverted this process and obtained probabilities from compression." [67, p. 79].

but there is nevertheless an asymmetry between IC and concepts of probability, as described in [100, Section 8.2]:

[19] "... 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?"

1. Absolute and conditional probabilities may be derived from SP-Multiple-Alignments within the SPTI (Section 6.3.12), but the SP-Patterns that match each other within that SP-Multiple-Alignment may not be derived from probabilities.

2. Structures such as words and phrases (Section 4.6.1), and 2D, 3D, and 4D structures (Section 6.4.7), and corresponding probabilities, may be derived via IC, but that kind of potential appears to be missing from most kinds of analysis of probabilities.

3. As described in [100, 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.

11.1.4 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.[20] But with ICMUP, as explained in [100, Section 8.2.3], the sizes of repeating SP-Patterns are as important as their frequency—which means that with matches between medium-to-large SP-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

[20] 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." [67, pp. 74–75].

New

0 (Other options for New are described in the text)

Old

X a 0 #X
X b X #X 1 #X

Figure 64: SP-Patterns corresponding to a PCS for the creation or recognition of unary numbers.

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 SP-Pattern and what is typically a much larger SP-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 ([89, Section 10.2], [86, Section 7.8]).

11.2 The potential of the SPTI as a theory of computing

This section considers the potential of the SPTI as a model of 'computing'. There are related discussions in [86, Chapter 4] and [100, Section 7].

11.2.1 With the SP Computer Model, the creation and recognition of numbers in unary notation

Figure 64 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 H.3, above).

The New SP-Pattern corresponds to the axiom in the PCS while the SP-Pattern '`X b X #X 1 #X`' is equivalent to the production. The SP-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 SP-Symbols '`X #X`' in the SP-Pattern '`X b X #X 1 #X`' may be read as 'any unary number' and the whole SP-Pattern may be read as 'a unary number is any unary number followed by 1', a recursive description much like the production '$ → $ 1'.

Given the SP-Pattern '0' in New and the other two SP-Patterns in Old, the SPCM creates a succession of good SP-Multiple-Alignments, one example of which is shown in Figure 65 (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 65 (a) into a single sequence and then ignore the 'service' SP-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.

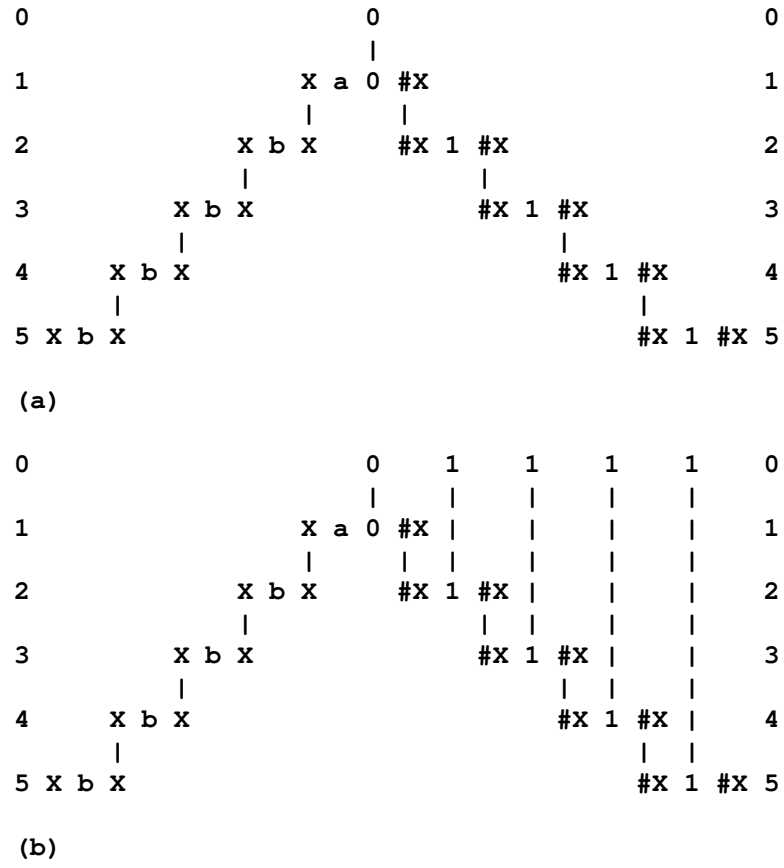

Figure 65: (a) One of many good SP-Multiple-Alignments produced by the SPCM with the SP-Pattern '0' in New and the Old SP-Patterns from Figure 64 in Old. (b) The best SP-Multiple-Alignment produced by the SPCM with '0 1 1 1 1' in New and the same SP-Patterns as with Figure 65 (a) in Old.

Figure 65 (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 may be run 'backwards' to recognise input SP-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 11.2.1 has described, with examples, how the operation of a PCS in normal form may be understood in terms of the SPTI. Since it is known that any PCS may be modelled by a PCS in normal form ([56], [50, 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 [50, Chapter 14] we may also conclude that the operation of any universal Turing machine may be interpreted in terms of the SPTI.

11.2.2 Potential benefits of the SPTI as a model of computing

Assuming the SPTI can be developed as outlined in Section H, whilst retaining its strengths in intelligence-related aspects of computing and beyond (Chapter 9), some of the potential benefits are outlined in the following subsections.

11.2.3 Gains in computational efficiency

In Chapter 4, we saw how, in brains and nervous systems, IC would be favoured by natural selection for the following reasons:

- Direct benefits from IC (Section 4.2):
  - 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 4.5).

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.

11.2.4 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 (Chapter 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 G.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.

11.2.5 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 (Chapter 11). For much the same reasons as were described in Section 11.2.4, this is not as absurd as it may seem.

11.2.6 Potential benefits of the SPTI as a model of computing

Assuming the SPTI can be developed as outlined in Section H, whilst retaining its strengths in intelligence-related aspects of computing and beyond (Chapter 9), some of the potential benefits are outlined in the following subsections.

11.2.7 Gains in computational efficiency

In Chapter 4, we saw how, in brains and nervous systems, IC would be favoured by natural selection for the following reasons:

- Direct benefits from IC (Section 4.2):
  - 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 4.5).

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.

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 (Chapter 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 Appendix G.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.

12 Conclusion

The aim of this book is to add substance to the main ideas in the SP Theory of Intelligence and its realisation in the SP Computer Model, and their applications. In brief:

- The SPTI has been developed with information compression (IC) at its core because of substantial evidence for the importance of IC in human learning, perception, and cognition (Chapter 4).

  Since the SPTI draws on evidence for the importance of IC in human learning, perception, 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, in general, IC may be achieved via the matching and unification of SP-Patterns (ICMUP, Appendix G), and more specifically via the concept of SP-Multiple-Alignment (Section 6.3).

- Paradoxical as this may sound, ICMUP and SP-Multiple-Alignment may achieve both the compression and decompression of information (Section 8.7).

- A central part of the SPTI is the SP-Multiple-Alignment concept (Section 6.3), a versatile means of compressing information and a 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 varied intelligence-related knowledge, in any combination (Chapter 9).

  As noted amongst the main unique selling points (Chapter 2) and Section 6.3, the *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!*.

- 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, perception, and cognition, it should not be surprising to find that much of mathematics, perhaps all of it, may be understood 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 the foundations of mathematics.

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 to create a New Mathematics (NM), with the benefits of AI and more than 2,000 years of thinking about mathematics. Potential benefits include: 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, and new thinking about concepts of computation, with potential benefits in both cases.

Comments and questions are very welcome.

As noted in Section 8.4, the SPTI, with its realisation in the SPCM, is far from complete and is best regarded as a *foundation* for the development of an AGI, not an AGI as it stands now. Further research is needed, much of it described in [53]. 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
ANN	Artificial Neural Network
DNN	Deep Neural Network
HLPC	Human Learning, Perception, and Cognition
IC	Information Compression
ICMUP	Information Compression via the Matching and Unification of Patterns
MSA	Multiple Sequence Alignment
NL	Natural Language
PCS	Post Canonical System
QM	Quantum Mechanics
TG	Transformational Grammar
SPCM	SP Computer Model
SPTI	SP Theory of Intelligence

Acknowledgements and funding

Developing the *SP Theory of Intelligence* (SPTI) and and its realisation in the *SP Computer Model* (SPCM) has not required any special provision of external resources or equipment. The main requirement has been time for reading, thinking, and the development and testing of the SPCM.

The most useful grant that I have been awarded was a Personal Research Grant HRP8240/1(A) to J. G. Wolff from the Social Science Research Council, 1984, when I was a lecturer at the University of Dundee. This freed me for a year to work full time on developing the SPTI and its realisation in the SPCM.

I'm grateful to Simon Tait, a colleague when I was employed at the software company Praxis Systems plc in Bath, England, for our lunch-time brainstorming sessions.

I'm grateful to the University Hospital of Wales for funding my research (into how a child learns his or her first language), both at work and in the University of Wales, Cardiff—research which was the basis for my PhD (summarised in [80]) and also led to the development of the SPTI.

I'm also grateful to Bangor University's School of Computer Science and Electronic Engineering in Bangor, Wales, for supporting the development of the SPTI

during working hours, and for the provision of an office and research facilities for a period of five years following my early retirement.

And I'm grateful to members of the 'SP Group' (https://groups.google.com/g/sp-research-news), especially Professor Gordana Dodig Crnkovic of Mälardalen University and Professor Vasile Palade of Coventry University, for their thinking and for practical and moral support. Gordana very kindly recommended this book to World Scientific, who accepted her recommendation.

Software availability

A ZIP file, 'SP71-2019.zip', containing the source code for the SP71 program (developed with Microsoft Visual C++), with examples of input and output files and some related files, may be downloaded from JGW's Google Drive via this link: https://tinyurl.com/467uxr9e which is valid for anyone with the link.

Clicking on the link above displays the name of the 'SP71-2019.zip' file with some details. To download the file, click on the download SP-Symbol near the top-right of the screen.

The file 'go.bat' within 'SP71-2019.zip' is a Windows executable file that simplifies the running of the program.

The source code for the SP71 program may also be downloaded from "Ancillary files" in the arXiv record: https://arxiv.org/abs/1306.3888 for "The SP theory of intelligence: an overview".

Appendix

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 Appendix G.

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 Chapter 4, Section 9.1, Section 9.2, 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 the name '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.*

SP-Multiple-Alignment The SP-Multiple-Alignment concept is described in Section 6.3.

SP-Symbol . 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 (Section 6.2).

SP-Pattern . An SP-Pattern is an array of *SP-Symbols* in one or two dimensions (Section 6.2)..

SP-Grammar . An SP-Grammar is a set of Old SP-Patterns that provides for the economical encoding of a relatively large body of New information (Section 6.4.1).

B Simplicity and Power

In accordance with Ockham's razor, a theory, derived via unsupervised learning from a body of data **I**, should be simple but not so simple that it says little or nothing that is useful. Here, Simplicity may be measured as $s = n_1 - n_2$ bits, where s is the value of Simplicity and n_1 is the size in bits of **I** before the given process of unsupervised learning and n_2 is the size of New after that process of information compression.

In the SPCM this learning 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**. Here, the size of Power may be seen to be $p = n_2$ bits.

B.1 Aim to make I as large as possible

Measures of Simplicity and Power are more important when they apply to a wide range of phenomena **I** than when they apply only to a small piece of data—because, for a given ratio of Simplity to Power, the absolute values for Simplicity or Power, or both of them, are likely to be greater when **I** is larger.

B.2 Comparing one system with another

In comparing one learning system with another, one may use the ratio of values for Power and Simplicity: p/s bits, or their absolute values.

B.3 'Dirty data'

There is discussion of issues relating to 'dirty data' in Section 6.4.6.

B.4 The main ideas in this book

The SPTI, and some of its potential benefits and application are described most fully in [86] and more briefly in [89]. 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 for the importance of IC across diverse aspects of intelligence in people and other animals ([99], Chapter 4). 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 to some extent in unsupervised learning [61,65], but with little or no recognition of the importance of IC in other aspects of intelligence.

  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 aspects of intelligence 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 (Section 9.1, and in other areas less closely-related to AI (Section 9.2).

  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 descriptive and explanatory Power (Appendix B).

  As noted amongst the USPs and **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.

- *Strengths of the SPTI* that deserve special mention include:

  - *The need for transparency in the organisation and workings of the SPTI* ([105, Section 10]). Unlike deep neural networks (DNNs), the SPCM provides an audit trail for all its workings, and there is transparency in its output and 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* ([92, Sections VII, VIII, and IX], [105, 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 6.4.5, [105, Section 6]). The SPTI framework of ideas suggests that remarkably simple principles govern the phenomena of generalisation, the correction of over- and under-generalisations.

  - *Reducing or eliminating the corrupting effect of errors in data* (Section 6.4.6.
  - *How to learn usable knowledge from a single exposure or experience* ([105, Section 7]). Like people and unlike DNNs, the SPTI can learn usable knowledge from a single exposure or experience.

- *Mathematics as information compression* (Chapter 10). A second *major discovery* is that **mathematics may be seen as a set of techniques for IC, and their application** [100]. 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 [100, 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 including in particular the potential for automation of mathematics via compression of data (Chapter 11).
  - Logic and computing may also be understood in terms of IC [100, 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 8.7).

- *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 8.2).

  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 [84, 105].

- *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 ([94], [86, 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 6.6.1).

  SP-Neural as it develops has a better chance than DNNs of reflecting the organisation and workings of real neural networks (Section 6.6.2).

- *Potential of the SPTI as a theory of probabilities.* Apart from its strengths as a theory of human learning, perception, and cognition and AI, the SPTI has potential as a theory of probabilities, with substantial potential benefits (Section 11.1).

- *Potential of the SPTI as a theory of computing.* Apart from its strengths as a theory of human learning, perception, and cognition and AI, the SPTI has potential as a theory of computing, with substantial potential benefits (Section 11.2).

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 [30], followed by Nick Bostrom's book on *Super-intelligence* [9], 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 super-intelligence, Bostrom 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." [9, p. 5].

and he quotes with approval what I. J. Good has to say:

> "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." [30, p. 33].

In short, super-intelligence 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, with intelligence that is far above human intelligence, will be called 'super-intelligence'.

C.3 Possible sources of super-intelligence

This section considers three possible sources of super-intelligence. A later section discusses how possible risks may be met.

C.3.1 Super-intelligence via IC

On the strength of evidence summarised in Chapter 6 and Section 9.1, the variety of capabilities that we call intelligence may be understood as IC, and only 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 Appendix 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 super-intelligent AI might lead to an intelligence explosion, as outlined in Appendix C.1.

C.3.2 Super-intelligence 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 Super-intelligence 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'.

Any AI may become super-intelligent via volume that, via compression of raw data, has been created by one or more other AIs or people. Provided it has enough memory, a floor-sweeping robot may be an instant expert on pre-raphaelite art, or civil engineering, and so on. The nearest equivalence for people is that they become expert via study or by consulting human experts.

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 For people, the risks of super-inrelligence

For people, the main risks of super-intelligence appear to be these:

- *Unemployment.* When super-intelligent AIs can do everything, there need be nothing for people to do. In that case, there would obviously be a need for everyone to have an income. We could all become the 'idle rich'.

- *Boredom.* But, like the idle rich, we might become bored, and since "The devil makes work for idle hands", some people at least might become destructive. To combat this kind of problem, we might ask the super-intelligent AIs,

in consultation with people, to provide entertainments which, for the more intelligent people, might include study.

- *Dominance*. A feature of AIs is that, like ordinary computers, they would have no intrinsic motivations except what their creators had given them.

The last point is not as reassuring as it might at first sight seem. If we lived in a world where AIs were entirely peaceful, without any desire to dominate people or other AIs (like Hitler or Pol Pot), then we could relax.

But it could easily happen that Bond villains or other people with those kinds of motivation would seek to create AIs to dominate people or other AIs or both, so that those human villains could attain their own sinister goals.

Solving this problem is at least as difficult as the problem of controlling nuclear weapons. The fact that, so far, we have avoided a nuclear holocaust suggests that we can, perhaps, keep control over super-intelligent AIs.

C.5 What can be done to avoid the potential risks of super-intelligence?

What follows is a few ideas about how the risks of super-intelligence may be reduced or eliminated.

C.5.1 Standardisation or limits?

There may be a case for some kind of standardisation of AIs, but 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.

Since there will be many uses for AIs that can compress information at high speed and AIs that can store large amounts of information, it seems unlikely that restrictions would be imposed on those kinds of AI.

C.5.2 Stop research in AI?

If we deliberately fail to reach human-level AI, risks from super-intelligent 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.5.3 Take advantage of zero motivations of computers?

As mentioned above, 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 super-intellient AI.

A potential problem here is that it is inevitable that any AI with human intelligence or above may 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.5.4 An international treaty to control motivations in super-intelligences?

The obvious weakness in the proposal in Appendix C.5.3 is that anyone, anywhere, may add motivation to a super-intelligent AI. Hence, the problem of super-intelligent AIs is not in the super-intelligence itself, it is largely the problem of preventing the addition of any motivation to any super-intelligence, or tightly controlling what kinds of motivation may be added.

In this connection, Kenan Malik writes:[21]

> "... 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."
> Kenan Malik, *The Observer*, 2023-11-26.

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 super-intelligence, or how people might exploit any super-intelligence, might allow us to gain the benefits of super-intelligence and to minimise the risks. Areas to be considered include:

[21] In 'AI doesn't cause harm by itself. We should worry about the people who control it', *The Observer*, 2023-11-26.

- Whether or not one may, safely, create super=intelligences 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 super=intelligences which do not have any motivations in themselves but may provide guidance for 'bad guys' with sinister motivations.

C.5.5 The need for transparency in the organisation, workings, and output of any super-intelligence

If it is accepted that the best way to minimise the risks of any super-intelligence would be via some kind of agreement or treaty to constrain what kinds of motivation or motivations, if any, may be added to any super-intelligence, or how any super-intelligence may be exploited by people, an issue that would require clarification would be how to assess any given super-intelligence, or proposed use of a super-intelligence, 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 super-intelligence, and transparency in its output. Without transparency in those aspects, there is potential for unwelcome motivations to be hidden from view within the super-intelligence, 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 super-intelligence:

- *Deep neural networks.* Any super-intelligence derived from DNNs may inherit weaknesses in that technology: lack of transparency in how DNNs work, and lack of transparency in their output.
- *The SPTI.* Any super-intelligence 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 [104].

D Solomonoff's development of Algorithmic Probability Theory (APT)

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 [66]. In a later paper [67], he describes useful background to the research and a relatively informal but more comprehensible description of the theory.

D.1 'Ad-Hoc' and 'Promiscuous' Grammars

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?" [67, p. 77].

Then he provides a preliminary answer:

> "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?" (*ibid.*).

After some detailed discussion, Solomonoff provides an informal summary of a key idea:

> "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." [67, p. 79].

and later again he quotes Andrey Kolmogorov:

> "He defined the algorithmic complexity of a string to be the length of the shortest code needed to describe it." [67, p. 83].

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. That method is described, with the method of calculating relative probabilities, in Section 6.3.12.

D.3 Finding good matches between two sequences of SP-Symbols

This appendix, based on part of [82],[22] 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 Appendix G.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.

D.3.1 The hit structure

Figure 66 illustrates the main concepts introduced in the description that follows. In this description, the query and the database are both sequences of atomic SP-Symbols, assumed to be characters in the discussion, and the database may be divided into sections such as sentences or paragraphs.

[22]Copyrigh © Sage Publications Ltd, 1994, reproduced by permission of Sage Publications Ltd.

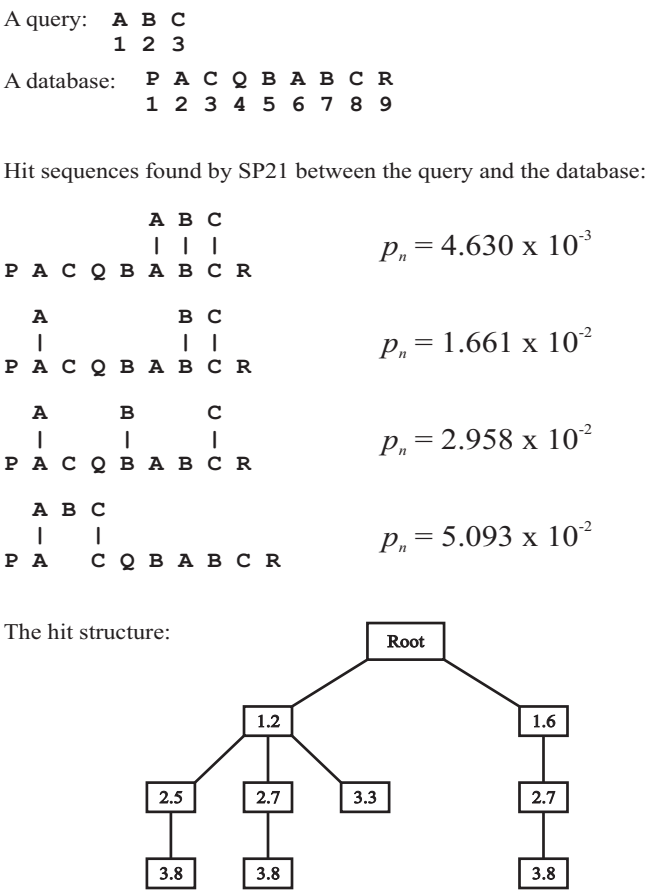

Figure 66: Concepts in pattern-matching and search. A 'query' string and a 'database' string are shown at the top with the ordinal positions of characters marked. Sequences of hits between the query and the database are shown in the middle with corresponding values of $p-n$ (described in the text). Each path from the root node to a leaf node represents a hit sequence. Reproduced with permission from [86, Figure A.1].

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) 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.

D.4 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 66. 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 SP-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.

D.5 Discussion of the search technique

The technique which has been described incorporates the principles of metrics-guided heuristic 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 thus 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 SP-Symbols is not in itself especially new and has been described elsewhere [12,43]. The novelty of the technique which has been described is in the way the broadcasting of SP-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 SP-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 [7].
- 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 SP-Patterns.

D.6 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.

D.6.1 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 SP-Symbols in the query match any of the SP-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 SP-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 SP-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.

D.6.2 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 $O(n \cdot m)$, where n is the number of characters in the query and m is the number of characters in the database.

D.6.3 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 $O(n \cdot m)$. 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.

D.6.4 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 $O(n)$.

D.6.5 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 $O(m)$.

D.6.6 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) [82]. 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 $O(n \cdot m)$.

E 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, perception, 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, redundancies of the kinds 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." [86, Section 2.3.7].

F 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 heuristic technique include hill climbing, heuristic gradient descent, 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. This kind of technique allows a New SP-Pattern (sometimes more than one) in row 0 (or column 0) to be encoded economically in terms Old SP-Patterns.

In the SPTI, this kind of strategy is normally required in the creation of two kinds of entity:

- In each SP-Multiple-Alignment, there is one Old SP-Pattern in each row (or column) above 0.

- In each SP-Grammar, there is a relatively large set of Old SP-Patterns that provides for the economical encoding of a relatively large body of New information.

G The working hypothesis that information compression may always be achieved via the full or partial matching and unification (merging) of patterns (ICMUP)

In view of the importance of IC in human learning, perception, and cognition (Chapter 4), 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 Chapter 4.

With regard to the evidence outlined in Chapter 4 (the importance of IC in HLPC), the phenomena considered in Sections 4.5 (IC and probabilities), and 4.6 (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 SP-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 SP-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, [21].

- To be clear, the concept of 'SP-Pattern' in this context includes single SP-Symbols. Thus there may be redundancy in a body of information, **I**, because some SP-Symbols occur more frequently than one would expect by chance, even though **I** does not contain any redundancy in the form of SP-Patterns with two or more SP-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 (Appendix G.2.7);

- The SP-Multiple-Alignment concept underpins the several intelligence-related strengths of the SPTI (Sections 9.1 and 9.2);

- And it appears that much of mathematics, perhaps all of it, may be understood in terms of ICMUP (Chapter 10).

G.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 [86, Section 2.2.8.4].

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

$$r = \sum_{i=1}^{i=n} (f_i - 1) \cdot s_i, \tag{13}$$

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 SP-Symbols, the number of possible subsequences (including single SP-Symbols and all composite patterns, both coherent and fragmented) is:

$$p = 2^m - 1. \tag{14}$$

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

$$c = p(p-1)/2. \tag{15}$$

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 F)—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 SP-Symbols in Appendix D.3.

G.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 [100], that, to a large extent, mathematics may be understood as ICMUP.

G.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 67 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 G.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 Appendix G, 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.

G.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 Appendix G.2.2 and described in more detail here:

- A unified pattern like '`INFORMATION`', which is often referred to as a 'chunk' of information,[23] is stored in a dictionary of patterns, as mentioned in Appendix G.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 67.

[23]There is a little more detail about the concept of 'chunk' in [99, Section 2.4.2].

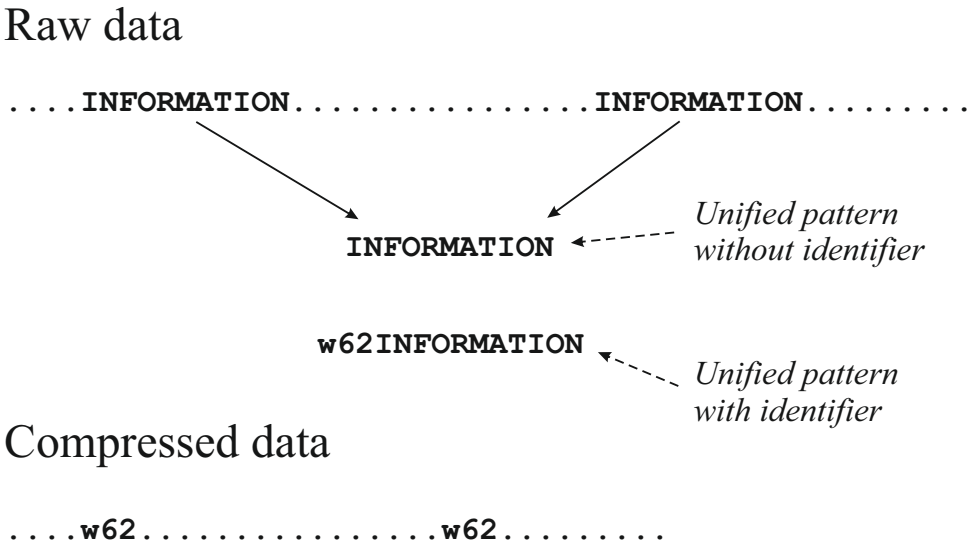

Figure 67: A schematic representation of the way two instances of the pattern '`INFORMATION`' in a body of data may be unified to form a single 'unified pattern', shown just above the middle of the figure. To achieve lossless compression, the relatively short identifier '`w62`' may be assigned to the 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 [86].

- 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 67.

- 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 (Appendix G.2.1).

G.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`'.

G.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:

'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.

G.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.[24]

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*.

[24]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 [8] and is now widely adopted in modern programming languages.

G.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.

G.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 [106].

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 9.1 and 9.2, described fairly fully in [89], and much more fully in [86] with many examples of SP-Multiple-Alignments.

G.2.8 An eighth compression technique via 'objects'

The concept of an 'object' in computer programming, introduced in the Simula language [8] 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 G.2.2) but introduces a third dimension additional to the one dimension of patterns in the SPCM as it is now, or the two dimensions of 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 [90, Sections 6.1 and 6.2].

G.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 [107, 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 (Chapter 9).

H Concepts of computing and the Post Canonical System (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' [71] and that other models of computing (such as 'Lambda Calculus' (Church and Kleene, see [57]), 'Recursive Function' [41], 'Normal Algorithm' [48] and Post's 'Canonical System' [56] are equivalent.

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 [72], 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 (Chapter 1) and in Section 8.4, 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 (Chapter 6, and Sections 6.3 and 6.4). 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.

H.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 *SP-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.[25] Each $\$\_i$ is an 'arbitrary' or 'variable' string, which can be null. Each $\$'\_i$ is to be replaced by a certain one of the $\$\_i$.' [50, 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 SP-Symbols, and both instances of '$' represent a single 'variable' which may have a 'value' comprising a string of zero or more SP-Symbols.

It has been proved [56] that any kind of PCS can be reduced to a PCS in normal form [50, Chapter 13]. That being so, a PCS in this form will be the main focus of our attention.

H.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 SP-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

[25] Spaces between SP-Symbols here and in other examples have been inserted for the sake of readability and because it allows us to use atomic SP-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.

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:

$$\begin{aligned} &\text{a \$} \rightarrow \text{\$ a} \\ &\text{b \$} \rightarrow \text{\$ b} \\ &\text{c \$} \rightarrow \text{\$ c.} \end{aligned}$$

In this example, the first SP-Symbol of the input string matches the first SP-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 SP-Symbol of 'b c d t a' matches the first SP-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 [50, 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.

H.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 SP-Symbols 0 and 1.

- Axiom: 0.
- Production: If any string '$' is a number, then so is the string '$ 1'.

  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 SP-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.

I Big tech companies, AI, and intellectual property

An article in the Guardian[26] says:

> "Sir Elton John has joined Sir Paul McCartney in calling for new rules to prevent tech companies from riding 'roughshod over the traditional copyright laws that protect artists livelihoods'. A petition against the unlicensed use of creative works for training generative AI now has more than 40,000 signatories including Julianne Moore, Kazuo Ishiguro, Kate Bush and Sir Simon Rattle. It is a battle that has united artists of every kind."

Since the UK Government is currently (June 2025) inclined to allow big tech companies to use copyright-protected creative works for training generative AI, this issue promises to run and run.

My own view is that copyright laws should be respected and that creative works should not be used for training generative AI; and that for the following reasons it is probably not sensible to consider possible schemes for compensating the authors of creative works:

[26] "The Guardian view on AI and copyright law: big tech must pay", Fri 31 Jan 2025 17.30 GMT

- ChatGPT and similar examples of 'generative AI' are not AI, they are merely statistical analyses of the human language used to train generative AI systems like ChatGPT.
- Thus the apparent intelligence of generative AI systems is human intelligence in a 'statistical parrot', not artificial intelligence.
- By allowing researchers to use copyright-protected creative works for training generative AI systems, the government would, in effect, be encouraging researchers to persist with research which would be an intellectual blind ally.
- With that blind ally closed, researchers may concentrate on the essentials of intelligence. Of course that includes the development of the SPTI, as outlined in [53], but many other avenues are possible.

J Barlows change of view about the significance of IC in mammalian learning, perception, and cognition, with comments

This section is reproduced with permission from [99, Appendix B].

As noted in Section 3.1.1, Horace Barlow has argued that:

> "... the [compression] idea was right in drawing attention to the importance of redundancy in sensory messages ... but it was wrong in emphasizing the main technical use for redundancy, which is compressive coding." [5, p. 242].

His main arguments follow, with my comments after each one, tagged with '`JGW`'.

> B.1. "*Redundancy is not something useless that can be stripped off and ignored.* It is important to realize that redundancy is not something useless that can be stripped off and and ignored. An animal must identify what is redundant in its sensory messages, for this can tell it about structure and statistical regularity in its environment that are important for its survival." [5, p. 243], and "It is ... knowledge and recognition of ... redundancy, not its reduction, that matters." [5, p. 244].

`JGW`: Barlow is right to say that knowledge of and recognition of redundancy is important "for this can tell [an animal] about structure and statistical regularity

in its environment that are important for its survival". In keeping with that remark, knowledge of the frequency of occurrence of any pattern may serve in the calculation of absolute and relative probabilities ([88, Section 3.7], [89, Section 4.4]) and it can be the key to the correction of errors, as Barlow mentions in the quote from him in the heading of Appendix B.2.

But, in the SP System, redundancy is not treated as "something useless that can be stripped of and ignored". Patterns that repeat are reduced to a single instance and the frequency of occurrence of that single instance is recorded. The existence of single instances like that, each with a record of its frequency of occurrence, is very important, both in the way that the SP System builds its model of the world and also in the way that it makes inferences and calculates probabilities of those inferences.

As noted in Section 10, if we did not compress sensory information, "our brains would quickly become cluttered with millions of copies of things that we see around us—people, furniture, cups, trees, and so on—and likewise for sounds and other sensory inputs". And, as noted in Section 3.1.1, Barlow himself has pointed out that the mismatch between the relatively large amounts of information falling on the retina and the relatively small transmission capacity of the optic nerve suggests that sensory information is likely to be compressed [3, p. 548]. And he has also pointed out that, in animals like cats, monkeys, and humans, "one obvious type of redundancy in the messages reaching the brain is the very nearly exact reduplication of one eye's message by the other eye" [4, p. 213], and because we normally see one view, not two, the duplication implies that the two views are merged and thus compressed. In general, the evidence presented in Sections 4 to 21 points strongly to IC as a prominent feature of HLPC.

> B.2. "*Redundancy is mainly useful for error avoidance and correction*" [5, p. 244].

`JGW`: The heading above, implies that compression of information via the reduction of redundancy is relatively unimportant, in keeping with the quotes from Barlow in the previous subsection.

Redundancy can certainly be useful in the avoidance of or correction of errors (Appendix C.2). But experience in the development and application of the SP Computer Model has shown that compression of information via the reduction of redundancy is also needed for such tasks as the parsing of natural language, pattern recognition, and grammatical inference. And compression of information may on occasion be intimately related to the correction of errors of omission, commission, and substitution, as described in Appendix C.2 and illustrated in Figure 19 (see also [89, Section 4.2.2] and [88, Section 6.2]).

B.3. "*There are very many more neurons at higher levels in the brain* and *Compressed, non-redundant, representation would not be at all suitable for the kinds of task that brains have to perform.*" [5, p. 244].

Following the remark that "This is the point on which my own opinion has changed most, partly in response to criticism and partly in response to new facts that have emerged." [5, p. 244].

Then Barlow writes:

> "Originally both Attneave and I strongly emphasized the economy that could be achieved by recoding sensory messages to take advantage of their redundancy, but two points have become clear since those early days. First, anatomical evidence shows that there are very many more neurons at higher levels in the brain, suggesting that redundancy does not decrease, but actually increases. Second, the obvious forms of compressed, non-redundant, representation would not be at all suitable for the kinds of task that brains have to perform with the information represented; ..." [5, pp. 244–245].
>
> and
>
> "I think one has to recognize that the information capacity of the higher representations is likely to be greater than that of the representation in the retina or optic nerve. If this is so, redundancy must increase, not decrease, because information cannot be created." [5, p. 245].

`JGW`: There seem to be two problems here:

(i) The likelihood that there are "very many more neurons at higher levels in the brain [than at the sensory levels]" and that "the information capacity of the higher representations is likely to be greater than that of the representation in the retina or optic nerve" need not invalidate ICHLPC. It seems likely that many of the neurons at higher levels are concerned with the storage of one's accumulated knowledge over the period from one's birth to one's current age ([88, Chapter 11], [94, Section 4]). By contrast, neurons at the sensory level would be concerned only with the processing of sensory information at any one time.

Although knowledge in one's long-term memory stores is likely to be highly compressed and only a partial record of one's experiences, it is likely, for most of one's life except early childhood, to be very much larger than the sensory information one is processing at any one time. Hence, it should be no surprise to find many more neurons at higher levels than at the sensory level.

(ii) For reasons given in Appendix B.4, next, there are reasons for doubting the proposition that "the obvious forms of compressed, nonredundant, representation

would not be at all suitable for the kinds of task that brains have to perform with the information represented."

B.4. *Compressed representations are unsuitable for the brain.*

Under the heading above, Barlow writes:

> "The typical result of a redundancy-reducing code would be to produce a distributed representation of the sensory input with a high activity ratio, in which many neurons are active simultaneously, and with high and nearly equal frequencies. It can be shown that, for one of the operations that is most essential in order to perform brain-like tasks, such high activity ratio distributed representations are not only inconvenient, but also grossly inefficient from a statistical viewpoint ..." [5, p. 245].

`JGW`: With regard to these points:

(i) It is not clear why Barlow should assume that a redundancy-reducing code would, typically, produce a distributed representation or that compressed representations are unsuitable for the brain. The SP System is dedicated to the creation of non-distributed compressed representations which work very well in several aspects of intelligence as outlined in Section 2.2.5 with pointers to where fuller information may be found. And in [94] it is argued that, in SP-Neural, such representations can be mapped on to plausible structures of neurons and their interconnections that are quite similar to Donald Hebb's [33] concept of a 'cell assembly'.

(ii) With regard to efficiency,

(a) It is true that deep learning in artificial neural networks [10], with their distributed representations, is often hungry for computing resources, with the implication that they are inefficient. But otherwise they are quite successful with certain kinds of task, and there appears to be scope for increasing their efficiencies [16].

(b) The SP System demonstrates that the compressed localist representations in the system are efficient and effective in a variety of kinds of task, as outlined in Section 2.2.5 with pointers to where fuller information may be found.

References

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

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

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

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

[5] H. B. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12:241–253, 2001.

[6] 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.

[7] A. A. Bertossi, F. Luccio, L. Pagli, and E. Lodi. A parallel solution to the approximate string matching problem. *Computer Journal*, 35(5):524–526, 1992.

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

[9] N. Bostrom. *Superintelligence*. Oxford University Press, Oxford, Kindle edition.

[10] M. Bramer. *Logic Programming With Prolog*. Springer-Verlag, London, second, Kindle edition, 2013.

[11] C. Brown. *My Left Foot*. Vintage Digital, London, Kindle edition, 2014. First published in 1954.

[12] 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*, 39(6):390–399, 1988.

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

[14] N. Chater. Reconciling simplicity and likelihood principles in perceptual organisation. *Psychological Review*, 103(3):566–581, 1996.

[15] N. Chater and P. Vitányi. Simplicity: a unifying principle in cognitive science? *TRENDS in Cognitive Sciences*, 7(1):19–22, 2003.

[16] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project Adam: building an efficient and scalable deep learning training system. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2014)*, pages 571–582. USENIX Association, 2014.

[17] N. Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[18] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.

[19] M. Chown. *The Magicians: Great Minds and the Central Miracle of Science*. Faber & Faber Ltd, London, Kindle edition, 2020.

[20] C. Coch. How the computer beat the go player. *Scientific American Mind*, 27(4):20–23, 2016.

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

[22] E. Davis and G. Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[23] H. L. Dreyfus. *What Computers Can't Do: A Critique of Artificial Reason*. Harper and Row, New York, 1972. Revised, 1979.

[24] T. G. Evans. A program for the solution of a class of geometric-analogy intelligence-test questions. In M. L. Minsky, editor, *Semantic Information Processing*, pages 271–353. MIT Press, Cambridge Mass., 1968.

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

[26] A. J. Gammerman. The representation and manipulation of the algorithmic probability measure for problem solving. *Annals of Mathematics and Artificial Intelligence*, 4:281–300, 1991.

[27] G. Gazdar and C. Mellish. *Natural Language Processing in Prolog*. Addison-Wesley, Wokingham, 1989.

[28] M. L. Ginsberg. AI and nonmonotonic reasoning. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3, pages 1–33. Oxford University Press, Oxford, 1994.

[29] M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[30] I. J. Good. Speculations concerning the first ultraintelligent machine. In F. L. Alt and M. Rubinoff, editors, *Advances in Computers*, volume 6, page 3188. Academic Press, New York, 1965.

[31] H. K. Hartline and F. Ratliff. Inhibitory interaction of receptor units in the eye of *limulus*. *Journal of General Physiology*, 40:357–376, 1957.

[32] H. K. Hartline, H. G. Wagner, and F. Ratliff. Inhibition in the eye of *limulus*. *Journal of General Physiology*, 39(5):651–673, 1956.

[33] D. O. Hebb. *The Organization of Behaviour*. John Wiley & Sons, New York, Kindle edition, 1949.

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

[35] A. S. Hsu, N. Chater, and P. Vitányi. Language learning from positive evidence, reconsidered: a simplicity-based approach. *Topics in Cognitive Science*, 5:35–55, 2013.

[36] D. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 49(9):1098–1101, 1952.

[37] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005. www.hutter1.net/ai/uaibook.htm.

[38] P. Jonas and G. Buzsaki. Neural inhibition. *Scholarpedia*, 2(9):3286, 2007.

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

[40] J. E. Kelly and S. Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press, New York, Kindle edition, 2013.

[41] S. C. Kleene. l-definability and recursiveness. *Duke Mathematical Journal*, 2:340–353, 1936.

[42] J. E. Laird. *The Soar Cognitive Architecture*. The MIT Press, Cambridge, Mass., 2012. ISBN-13: 978-0-262-12296-2.

[43] D. L. Lee. ALTEP - a cellular processor for high-speed pattern matching. *New Generation Computing*, 4(3):225–244, 1986.

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

[45] J. Lyons. *Introduction to Theoretical Linguistics*. Cambridge University Press, Cambridge, 1968.

[46] Phil Maguire, Oisín Mulhall, Rebecca Maguire, and Jessica Taylor. Compressionism: a theory of mind based on data compression. In *Proceedings of the 11th International Conference on Cognitive Science*, pages 294–299, 2015.

[47] G. F. Marcus and E. Davis. *Rebooting AI: Building Artificial Intelligence We Can Trust*. Pantheon Books, New York, Kindle edition, 2019.

[48] A. A. Markov and N. M. Nagorny. *The Theory of Algorithms*. Kluwer, Dordrecht, 1988.

[49] P. McCorduck. *Machines Who Think: a Personal Inquiry Into the History and Prospects of Artificial Intelligence*. A. K. Peters Ltd, Natick, MA, second edition, 2004.

[50] M. L. Minsky. *Computation, Finite and Infinite Machines*. Prentice Hall, Englewood Cliffs, NJ., 1967.

[51] A. Newell. You can't play 20 questions with nature and win: Projective comments on the papers in this symposium. In W. G. Chase, editor, *Visual Information Processing*, pages 283–308. Academic Press, New York, 1973.

[52] A. Newell, editor. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Mass., 1990.

[53] V. Palade and J. G. Wolff. A roadmap for the development of the 'SP Machine' for artificial intelligence. *The Computer Journal*, 62:1584–1604, 2019. `https://tinyurl.com/bdht5cpm`, `http://bit.ly/2tWb88M`, arXiv:1707.00614.

[54] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, revised second printing edition, 1997.

[55] 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*, 13:231–278, 1980.

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

[57] J. B. Rosser. Highlights of the history of the lamda-calculus. *Annals of the History of Computing (USA)*, 6(4):337–349, 1984.

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

[59] A. Roy. An extension of the localist representation theory: grandmother cells are also widely used in the brain. *Frontiers in Psychology*, 4:1–3, 2013.

[60] J. Schmidhuber. Driven by compression progress: a simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In G. Pezzulo, M. V. Butz, O. Sigaud, and G. Baldassarre, editors, *Anticipatory Behavior in Adaptive Learning Systems, from Sensorimotor to Higher-level Cognitive Capabilities*, Lecture Notes in Artificial Intelligence. Springer, Berlin, 2009.

[61] J. Schmidhuber. Deep learning in neural networks: an overview. *Neural Networks*, 61:85–117, 2015.

[62] J. Schmidhuber. One big net for everything. Technical report, The Swiss AI Lab, IDSIA, 2018. Download link: `https://arxiv.org/pdf/1802.08864`.

[63] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

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

[65] R. Shwartz-Ziv and Y. LeCun. To compress or not to compress—self-supervised learning and information theory: a review. *Entropy*, 26(3):1–39, 2024. arXiv:2304.09355v3 [cs.LG], `https://doi.org/10.3390/e26030252`.

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

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

[68] L. R. Squire, D. Berg, F. E. Bloom, S. du Lac, A. Ghosh, and N. C. Spitzer, editors. *Fundamental Neuroscience*. Elsevier, Amsterdam, fourth edition, 2013.

[69] Y. Tian and Y. Zhu. Better computer go player with neural network and long-term prediction. 2016.

[70] K. Tunyasuvunakool, J. Adler, Z. Wu, T. Green, M. Zielinski, A. dek, A. Bridgland, A. Cowie, C. Meyer, A. Laydon, S. Velankar, G. J. Kleywegt, A. Bateman, R. Evans, A. Pritzel, M. Figurnov, O. Ronneberger, R. Bates, S. A. A. Kohl, A. Potapenko, A. J. Ballard, B. Romera-Paredes, S. Nikolov, R. Jain, and D. Hassabis.

[71] A. M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265 and 544–546, 1936.

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

[73] M. Varadi, S. Anyango, M. Deshpande, S. Nair1, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Zidek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar.

[74] S. Watanabe. Information-theoretical aspects of inductive and deductive inference. *IBM Journal of Research and Development*, 4:208–231, 1960.

[75] S. Watanabe. Pattern recognition as information compression. In S. Watanabe, editor, *Frontiers of Pattern Recognition*. Academic Press, New York, 1972.

[76] E. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics*, 13:1–14, 1960. Richard Courant Lecture in Mathematical Sciences delivered at New York University, May 11, 1959.

[77] J. G. Wolff. Language acquisition and the discovery of phrase structure. *Language & Speech*, 23:255–269, 1980.

[78] J. G. Wolff. Language acquisition, data compression, and generalization. *Language & Communication*, 2(1):57–89, 1982.

[79] J. G. Wolff. Cognitive development as optimization. In L. Bolc, editor, *Computational Models of Learning*, pages 161–205. Springer-Verlag, Heidelberg, 1987.

[80] J. G. Wolff. Learning syntax and meanings through optimization and distributional analysis. In Y. Levy, I. M. Schlesinger, and M. D. S. Braine, editors, *Categories and Processes in Language Acquisition*, pages 179–215. Lawrence Erlbaum, Hillsdale, NJ, 1988. bit.ly/ZIGjyc.

[81] J. G. Wolff. Computing, cognition and information compression. *AI Communications*, 6(2):107–127, 1993. bit.ly/XL359b.

[82] J. G. Wolff. A scaleable technique for best-match retrieval of sequential information using metrics-guided search. *Journal of Information Science*, 20(1):16–28, 1994. See `www.cognitionresearch.org/papers/ir/ir.htm`.

[83] J. G. Wolff. Probabilistic reasoning as information compression by multiple alignment, unification and search. Technical report, School of Informatics, University of Wales at Bangor, 1998. See `www.cognitionresearch.org/papers/pr/pr.htm`.

[84] J. G. Wolff. The SP Theory of Intelligence, and its realisation in the SP Computer Model as a foundation for the development of artificial general intelligence. *Analytics*, 2(1):163–192, 20023. Web: www.mdpi.com/2813-2203/2/1/10.

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

[86] J. G. Wolff. *Unifying Computing and Cognition: the SP Theory and Its Applications*. CognitionResearch.org, Menai Bridge, 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.

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

[88] J. G. Wolff. Computing as compression: the SP Theory of Intelligence. 2013. bit.ly/113Z8eG.

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

[90] J. G. Wolff. Application of the SP Theory of Intelligence to the understanding of natural vision and the development of computer vision. *SpringerPlus*, 3(1):552–570, 2014.

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

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

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

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

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

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

[97] J. G. Wolff. Interpreting Winograd Schemas via the SP Theory of Intelligence and its realisation in the SP Computer Model. Technical report, CognitionResearch.org, 2018. bit.ly/2ME8DOA.

[98] J. G. Wolff. Commonsense reasoning, commonsense knowledge, and the SP Theory of Intelligence. Technical report, CognitionResearch.org, 2019. viXra:1901.0051v2, hal-01970147 version 3, bit.ly/2RESeut.

[99] J. G. Wolff. Information compression as a unifying principle in human learning, perception, and cognition. *Complexity*, 2019:1–38, February 2019. Download link: `https://tinyurl.com/bdd5jdxd`.

[100] J. G. Wolff. Mathematics as information compression via the matching and unification of patterns. *Complexity*, 2019:25, 2019. Download link: `https://tinyurl.com/3yx9nhnb`, This paper is reproduced in *New Ideas Concerning Science and Technology*, Vol. 13, 12 June 2021, Pages 132–169, `https://doi.org/10.9734/bpi/nicst/v13/8449D`.

[101] J. G. Wolff. How the SP System may promote sustainability in energy consumption in IT systems. *Sustainability*, 13(8):1–21, 2021.

[102] J. G. Wolff. The potential of the SP System in machine learning and data analysis for image processing. *Big Data and Cognitive Computing*, 5(1):7, 2021.

[103] J. G. Wolff. A proposed solution to problems in learning the knowledge needed by self-driving vehicles. Technical report, CognitionResearch.org, 2021. tinyurl.com/n6vpxcuf.

[104] J. G. Wolff. Transparency and granularity in the SP Theory of Intelligence and its realisation in the SP Computer Model. In Witold Pedrycz and Shyi-Ming Chen, editors, *Interpretable Artificial Intelligence: A Perspective of Granular Computing*. Springer, Heidelberg, 2021. ISBN 978-3-030-64948-7, arXiv:2009.06370 [cs.AI].

[105] 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*, 2:1045–1079, 2022. `https://doi.org/10.3390/foundations2040070`.

[106] 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*, 7(4):246–260, 2023.

[107] N. S. Yanofsky and M. Zelcer. The role of symmetry in mathematics. 2016. arXiv:1502.07803v2 [math.HO].