PROBABILISTIC REASONING AS ICMAUS (II): CALCULATION OF PROBABILITIES, BEST-MATCH PATTERN RECOGNITION AND INFORMATION RETRIEVAL, AND REASONING WITH NETWORKS, TREES AND RULES

J Gerard Wolff

University of Wales, Bangor, UK
gerry@sees.bangor.ac.uk

Abstract: This article and its two accompanying articles introduce the idea that probabilistic reasoning (PrbRs) may be understood as information compression by multiple alignment, unification and search (ICMAUS).

In the context of the first article which introduced the ICMAUS concepts and how they may be used to form inferences, this second articles shows, first, how the probabilities of inferences may be calculated.

The remainder of the articles presents examples showing how the ICMAUS framework may be used to model a selection of different kinds of PrbRs including:

- PrbRs associated with probabilistic and fuzzy pattern recognition and information retrieval. Associated topics include inheritance of attributes in a class hierarchy or heterarchy and the recognition of 'polythetic' categories.
- One-step 'deductive' reasoning.
- The integration of 'deductive' and 'abductive' reasoning within a single framework.
- Chains of reasoning:
    - Reasoning with networks or trees (but excluding Bayesian networks).
    - Reasoning with 'rules'.

Key Words: Probabilistic reasoning; fuzzy pattern recognition; best-match information retrieval; abductive reasoning; class hierarchy; inheritance; decision network; decision tree; rules;

Category: Computing Methodologies/Artificial Intelligence/Reasoning (uncertainty, 'fuzzy' and probabilistic). I.2.SD.1.2.3.

1 Introduction

This is the second of three articles [2, 3, 4] introducing probabilistic reasoning (PrbRs) as information compression by multiple alignment, unification and search (ICMAUS). This article assumes that the first article [2] has been read and that the reader understands, at least in outline, the concept of multiple alignment as it has been developed in this research. At least in broad-brush terms, the reader should also understand how the information compression (IC) associated with an alignment may be calculated and how probabilistic inferences may be drawn from an alignment.

This article shows, first, how the probabilities of inferences associated with any alignment may be calculated (Section 2). The remainder of the article presents a range of examples showing how different types or aspects of PrbRs may be modelled as ICMAUS. Areas covered include:

1. Probabilistic inferences associated with probabilistic and fuzzy pattern recognition and information retrieval (Section 3). Examples include:
    - Recognition of simple patterns (Section 3.2).
    - Patterns with internal structure exemplified by disease patterns used for medical diagnosis (Section 3.3).
    - Inheritance of attributes in a class hierarchy or heterarchy (Section 3.4).
    - Recognition of polythetic categories (Section 3.5).
2. Reasoning with networks, trees and rules (Section 4), including:
    - One-step 'deductive' reasoning (Section 4.1).
    - The integration of 'deductive' and 'abductive' reasoning within a single framework (Section 4.2).
    - Chains of reasoning:
        - Modelling the structure and function of a decision network or decision tree (excluding Bayesian networks) (Section 4.3).
        - Modelling the operation of 'rules' in an expert system (Section 4.4).

The majority of examples in this article show results produced by the SP61 model, which was described in Section 6 of [2]

2 Calculation of the probabilities of inferences

This section describes how the probabilities of ICMAUS inferences may be calculated. Although the example calculation is simple, the method to be described is general and may be applied to alignments of any complexity.

2.1 Frequencies of patterns

All the calculations described in this section are based on the frequency values associated with the patterns in Old. Before we proceed to describe the calculations, a few words are necessary about the status of the frequency values of patterns and symbols:

- In the case of patterns representing phrases and sentences in a natural language, and also 'function' words in natural language like 'the' or 'because', it is reasonably clear that the frequencies of these patterns are values which are measured or at least guestimated for the appearance of each pattern in some more or less representative sample or samples of the given language.
- Since syntactic patterns above the level of 'word' (and semantic patterns) do not normally appear directly in language samples, the frequency of any such pattern must be the frequency with which it is identified in a (good) parsing of the sample.
- In the case of words in natural language which represent entities in the world (eg 'table', 'house' etc), or patterns in a database which represent external entities (e.g., descriptions of human diseases or species of animal or plant), it is pertinent to ask whether the relevant frequency is the frequency with which the pattern is used or

whether it is the frequency with which the entity represented by the pattern appears in the world.

With regard to the last point, we can see that the two kinds of frequency are not always in step if we consider words or other patterns representing the giant panda and words or other patterns representing, say, beetles. The frequency of occurrence of giant pandas in the world is exceedingly small compared with the frequency of occurrence of beetles. But in terms of the frequency of representative words or other patterns, the giant panda probably wins, certainly in ‘popular’ sources and, possibly, in technical ones too.

Which of these two frequencies we use seems to vary, depending on the kind of inference that is required:

- If our interests are primarily concerned with language or other medium for representation, then we should use frequencies of words, phrases and sentences in a relevant sample of language. For example, we may wish to repair some damaged text by inferring letters or words to fill gaps or eliminate errors of insertion or substitution.
- If our interests lie primarily in the external world, then the second kind of frequency is more appropriate. For example, a system for diagnosing faults in cars (e.g., the one described in Section 4.2) needs to be conditioned by the frequencies with which different kinds of fault occur. If it is known that the engine failing to start is caused more often by a flat battery than by a failed starting motor, this information obviously has a bearing on the order in which we which we apply diagnostic tests and the probabilities which should attach to inferences formed by the system.

2.2 Absolute probabilities of alignments and inferences

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

$$N = |A|^L.$$

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

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

If we know the length of the code derived from an alignment, we can use this formula to calculate the probability of the New sequence in the alignment after it has been compressed in accordance with the alignment (or, if New is not completely matched, we can calculate the probability of that substring or subsequence within New which is matched to symbols in Old within the alignment).

This probability may also be regarded as a probability of the alignment and of the inferences contained in the alignment.

Before we apply this formula, there are two questions which need to be answered.

2.3 Is it reasonable to assume that the code for an alignment is a random sequence or nearly so?

In accordance with Algorithmic Information Theory (references in Section 1.4 of [2]), 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.

In this connection, there is a snag: in all but the simplest cases, it is not possible to prove for any given body of data that no further compression is possible. All we can do is to say that, with the methods we are currently using, and the resources we have applied, no further compression can be achieved.

Thus the assumption that the code for an alignment is random or nearly so only applies to the best codes found for a given body of information in New and must be qualified by the quality and thoroughness of the methods which have been used to create the code.

2.4 What are the values of L and |A|?

If we regard the code for an alignment as a sequence of symbol types drawn (with replacement) from those found in the corpus (with the possible addition of symbol types created by the system during ‘learning’) then the value of L is the length of the code in symbols and |A| is the number of symbol types just referred to.

However, for the reasons explained in Section 5 of [2], a stop-gap method is currently being used in SP61 which delivers a code length in bits. For this reason, the value of L must be the length of the code in bits (B_E in Section 5.5 of [2]) and the value of |A| is 2.

These values of L and |A| are not entirely satisfactory because, in general, the model operates with symbols which are larger than a single bit and, in general, the value of |A| is bigger than 2. However, for reasons given in Section 2.5, next, systematic biases in the calculated values of absolute probabilities which arise from these approximations may have little significance in practice.

2.5 Relative probabilities: alignments

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

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

If we are to compare one alignment and its probability to another alignment and its probability, ***we need to compare like with like***. An alignment can have a high value for p_{ABS} because it encodes only one or two symbols from New.[1] It is not reasonable to compare an alignment like that to another alignment which has a lower value for p_{ABS} but which encodes more symbols from New. Consequently, the procedure for calculating relative values for probabilities (p_{REL}) is as follows:

1 For the alignment which has the highest compression difference (which we shall call the *reference alignment*), identify the symbols from New which are encoded by the alignment. We may call these symbols the *reference set* of symbols in New.

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

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

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

$$p_{A\_SUM} = \sum_{i=1}^{R} p_{ABSi}$$

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

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

$$p_{RELi} = p_{ABSi} / p_{A\_SUM}.$$

The values of p_{REL} calculated as just described seem to provide an effective means of comparing the alignments which encode the same set of symbols from New as the alignment which has the best overall compression difference.

It is not necessary always to use the alignment with the best compression difference as the basis of the reference set of symbols. It may happen that some other set of symbols from New is the focus of interest. In this case a different reference set of alignments may be constructed and relative values for those alignments may be calculated as described above.

---

1. As was noted in [2], the IC associated with any alignment is not realised until New has been encoded in terms of the patterns from Old which appear in the alignment. However, it is convenient at times to say that a given alignment encodes New or compresses New. Wherever this form of words is used it should be understood to mean that a code which expresses New in a compressed form may be derived from the alignment under consideration.

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

3. If Old is well compressed, this kind of redundancy amongst the rows of an alignment should not appear very often.

2.6 Relative probabilities: patterns and symbols

It often happens that a given pattern from Old or a given symbol type within patterns from Old appears in more than one of the alignments in the reference set. In cases like these one would expect the relative probability of the pattern or symbol type to be higher than if it appeared in only one alignment. To take account of this kind of situation, SP61 calculates relative probabilities for individual patterns and symbol types in the following way:

1. Compile a set of patterns from Old, each of which appears at least once in the reference set of alignments.
2. For each pattern, calculate a value for its relative probability as the sum of the p_{REL} values for the alignments in which it appears. If a pattern appears more than once in an alignment, it is only counted once for that alignment.
3. Compile a set of symbol types which appear anywhere in the patterns identified in 2.
4. For each symbol type identified in 3, calculate its relative probability as the sum of the relative probabilities of the patterns in which it appears. If it appears more than once in a given pattern, it is only counted once.

With regard to symbol types, the foregoing applies only to symbol types which do not appear in New. Any symbol type which appears in New necessarily has a probability of 1 - because it has been observed, not inferred.

2.7 A simple example

In order to illustrate the kinds of values which may be calculated for absolute and relative probabilities, this subsection presents a very simple example, similar to the one mentioned earlier about the inference of '`fire`' from '`smoke`'. Here, we shall extend the concept of '`smoke`' to include anything, like mist or fog, which looks like smoke. To avoid confusion, we shall use the symbol '`a_smoke`' for anything which has the appearance of smoke. Also, '`fire`' has been divided into three categories: the kind of fire used to heat a house or other building, dangerous fires that need a fire extinguisher or more, and the kind of fire inside a burning cigarette or pipe.

Given the small knowledge base shown in Table 1 as Old, and a pattern containing the single symbol '`a_smoke`' as New, the frequency of occurrence and the minimum cost of each symbol type (calculated by SP61 using the method described in Section 5.1 of [2]) are shown in Table 2.

Patterns	*Frequency*	*Encoding cost (bits)*
`clouds black rain`	15,000	2.93
`dangerous fire a_smoke`	500	7.84
`heating fire a_smoke`	7,000	4.03
`tobacco fire a_smoke`	10,000	3.52
`fog a_smoke`	2,000	5.84
`stage a_smoke`	100	10.16
`thunder lightning`	5,000	4.52
`strawberries cream`	1,500	6.26

Table 1
A small knowledge base of associations. The middle column shows an imaginary frequency of occurrence of each pattern in some notional reference environment. The right-hand column shows the encoding cost of each pattern, calculated as described in Section 5.2 of [2].

The encoding cost of each pattern is simply the sum of the minimum costs of each of the minimum number of symbols from the pattern which are needed to discriminate the pattern uniquely within Old (as described in Section 5.2 of [2]). In this example, every pattern can be identified uniquely by its first symbol. Thus, in all cases, the encoding cost of each pattern (shown to the right of Table 1) is the minimum cost of its first symbol.

	Frequency	*Minimum cost (bits)*
`a_smoke`	19,601	2.55
`black`	15,000	2.93
`clouds`	15,000	2.93
`cream`	1,500	6.26
`dangerous`	500	7.84
`fire`	17,500	2.71
`fog`	2,000	5.84
`heating`	7,000	4.03
`lightning`	5,000	4.52
`rain`	15,000	2.93
`stage`	100	10.16
`strawberries`	1,500	6.26
`thunder`	5,000	4.52
`tobacco`	10,000	3.52

Table 2
The frequency of occurrence of each symbol type appearing in Table 1 (together with New which is a pattern containing the single symbol '`a_smoke`') and the minimum cost of each symbol type calculated by the method described in Section 5.1 of [2].

Given that New is a pattern containing the single symbol '`a_smoke`', SP61 forms the five obvious alignments of New with each of the patterns in Table 1 which contain the symbol '`a_smoke`'. The absolute and relative probabilities of the five alignments, calculated as described above, are shown in Table 3.

	Absolute probability	*Relative probability*
a_smoke/tobacco fire a_smoke	0.08718	0.51020
a_smoke/heating fire a_smoke	0.06103	0.35714
a_smoke/fog a_smoke	0.01744	0.10204
`a_smoke/dangerous fire a_smoke`	0.00436	0.02551
a_smoke/stage a_smoke	0.00009	0.00510

Table 3

Absolute and relative probabilities of each of the five reference alignments formed between '`a_smoke`' in New and, in Old, the patterns shown in Table 1. The relative probability of each pattern from Old is the same as the alignment in which it appears.

In this very simple example, the relative probability of each pattern from Old is the same as for the alignment in which it appears. However, the same cannot be said of individual symbol types. The relative probabilities of the symbol types that appear in any of the five reference alignments (shown in Table 3) are shown in Table 4.

	Relative probability
`a_smoke`	1.00000
`fire`	0.89286
`tobacco`	0.51020
`heating`	0.35714
`fog`	0.10204
`dangerous`	0.02551
`stage`	0.00510

Table 4

The relative probabilities of the symbol types from Old that appear in any of the reference set of alignments shown in Table 3.

The main points to notice about the relative probabilities shown in Table 4 are:

- The relative probability of '`a_smoke`' is 1.0. This is because it is a 'fact' which appears in New: hence there is no uncertainty attaching to it.
- Of the other symbol types from Old, the one with the highest probability relative to the other symbols is '`fire`', and this relative probability is higher than the relative probability of any of the patterns from Old (Table 3). This is because '`fire`' appears in three of the reference alignments.

In this example, we have ignored all the subtle cues that people would use in practice to infer the origin of smoke: the smell, colour and volume of smoke, associated noises, behaviour of other people, and so on. Allowing for this, and allowing for the probable inaccuracy of the frequency values which have been used, the relative probabilities of alignments, patterns and symbols seem to reflect the subjective probability which we might assign to the five alternative sources of smoke-like matter in everyday situations. Our general intuition is that if we see smoke (or something that looks like smoke) we are likely also to find some kind of fire. If we are more discriminating in our inferences, we may guess that, despite falling numbers of smokers in some countries, burning tobacco probably generates more smoke in our ordinary experience than the kind of fire used for heating or which needs a fire extinguisher. Generally, we are aware that something that looks like smoke may in fact be fog or mist. In most situations except the theatre, the kind of smoke which is used on stage would have a low subjective probability.

2.8 Comparison of alignments which do not encode the same symbols from New

It is true that, when we compare alignments, we should compare like with like in terms of the symbols from New which are encoded by the alignment. But, nevertheless, there may be occasions when we wish to compare alignments which do not encode the same symbols from New. It might happen, for example, that each of the best five alignments for a given combination of New and Old encodes a subsequence of the symbols from New which is different from any other alignment but which overlaps with subsequences for other alignments.

2.8.1 Using ‘compression ratio’

In a case like this, it may be useful to calculate a ‘compression ratio’ for each alignment as:

$$CR = (B_E / B_N) \times 100$$

which may be interpreted as a measure of the number of bits of code required to encode 100 bits of New. It should also be able to calculate values for p_{ABS} in the same way as before. Then any set of the alignments for the given combination of New and Old may constitute a reference set and we can calculate relative probabilities for alignments, patterns and symbol types as before.

This way of calculating relative probabilities may be useful when the best alignments encode overlapping subsets of the symbols in New and none of them encode all the symbols in New.

The above method for calculating relative probabilities for patterns and symbols does not seem entirely satisfactory because, where a pattern or symbol type occurs two or more times in an alignment, it is only counted once for that alignment. There seems to be a need to discriminate amongst appearances of patterns or symbol types according to the context in which they occur and to calculate relative probabilities for each entity in a given context.

2.8.2 Combining alignments

An alternative to the proposal in the previous subsection is to combine alignments so that, together, they encode the same symbols from New as the alignments in the reference set of the best alignment. For example, the best alignment and the other alignments in its reference set may encode all the symbols in the pattern ‘`A B C`’ in New. However, there may be three other alignments, each of which encodes only one of the symbols in ‘`A B C`’ but which, together, encode all three symbols.

In a case like this, it seems that one can construct an alignment as a concatonation of the three alignments just mentioned. An absolute probability can be calculated for this composite alignment in two possible ways:

- The composite alignment may be processed in exactly the same way as any other alignment and the absolute probability may be calculated as before.
- A possible short-cut is simply to multiply the absolute probabilities of the alignments from which the composite alignment was formed.

After a value has been obtained for the absolute probability of the composite alignment, the alignment may be added to the reference set of alignments and processed alongside the other alignments in the reference set as before.

What happens if we have an alignment that encodes '`A B`' and another that encodes '`B C`'? It seems that, in a manner similar to the previous case, these two alignments may be combined to form a single alignment that encodes all three symbols from New and this combined alignment may be added to the reference set of alignments and processed alongside the other alignments in the reference set.

In general, alignments which are constructed by concatonating or combining other alignments are likely to contain more rows and more patterns from Old than the original alignments in the reference set and this will mean that the absolute and relative probabilities associated with composite alignments are likely to be lower than for the original alignments.

2.9 Confidence in inferences

The formulae and calculations presented in this section may suggest that, given a system which is equipped with this mathematical machinery, we can calculate precisely the level of confidence that should be placed in inferences which the system makes.

This is, of course, nonsense. Any knowledge-based system (including human brains!) is subject to the law "rubbish in means rubbish out". In the case of human experts, we do or should ask whether they were given all the relevant information about a given case, how well-trained they are, how much experience they have, whether they have given full attention to the case, and so on. Artificial systems are no different.

In general, the confidence which we may place in probabilistic inferences made by a given system should be influenced by several factors including the accuracy and coverage of the information supplied about a particular case, the accuracy and coverage of the knowledge stored in the system, the effectiveness of the search methods used by the system, and the thoroughness of the search which has been made for a given set of data and inferences.

3 Best-match pattern recognition and information retrieval as ICMAUS

This section considers how pattern recognition may be modelled in the ICMAUS framework. In this context, the term *pattern recognition* (PtR) mean the matching of 'input' information against stored patterns so that good partial matches can be identified as well as exact matches. As was discussed in Section 7.1 of [2], identification of good partial matches is a manifestation of PrbRs because any such match contains an implicit inference that the unmatched parts of the stored pattern are present or should be present together with the parts of the pattern which have been matched - and these inferences are, in general, probabilistic.

Any of the examples of pattern recognition described in this section my also be regarded as an example of *information retrieval* (IR). However, there are kinds of IR which would not normally be regarded as PtR. An example of one such kind of IR is described in the third of these three articles [4].

Of course, PtR and IR are both large subjects which cannot be discussed exhaustively in the available space. The subsections which follow attempt merely to sketch some of the ways in which the ICMAUS framework may model some aspects of these topics.

3.1 Representation of numerical information

Before we proceed to consider the examples in this and later sections, one issue needs clarification: is it possible to represent numerical information in the ICMAUS scheme and, if so, how?

For the sake of simplicity in the model which is being developed, the principle has been adopted that, as far as possible, there should be no symbols in Old or New which have 'hidden' meaning. Every symbol should be merely a 'mark' which can be matched in a yes/no manner with other marks (see Appendix A1 of [2]). As a working hypothesis, it is assumed that any and every meaning which one might wish to be associated with a symbol or pattern of symbols may be represented by the context of other symbols in which the given symbol or pattern appears.

Given this principle, how can we accommodate arithmetic and other areas of mathematics where most symbols, e.g. '6', '10.2', '+' etc, have a mathematical meaning which is normally hidden from view? Of course, we are perfectly at liberty to use such symbols merely as 'marks' with no hidden meaning (and this is often convenient), but, without some appropriate provision, we cannot use them with their normal mathematical meaning.

3.1.1 Representing mathematical axioms and definitions with patterns

One answer to this question is to represent mathematical axioms and definitions in the form of patterns and symbols and to include such patterns in every knowledge base which requires mathematics. A working hypothesis in this research programme is that it will prove possible to represent all such meanings in this way, and preliminary work in this area lends support to this hypothesis.

However, research in this area has not proceeded far enough yet to make this a practical proposition right now. For the time being, something else is needed as a stop-gap pending a 'complete' solution to the problem.

3.1.2 A temporary solution

Pending something better, numerical information will be represented with symbols without mathematical meanings. For example, in a medical context (see Section 3.3), the temperature of a patient may be represented as ‘high’, ‘normal’ or ‘low’, or with different symbols or a different number of categories.

For many purposes, this kind of representation is quite adequate. Within the limits imposed by the granularity of the scale, it is possible to say whether two temperatures are ‘equal’ or ‘not equal’ but it not possible to compare values with ‘greater than’ or ‘less than’ or to use them in some other mathematical way.

3.2 Simple patterns

The first and simplest example of best-match PtR or IR which we will consider is where an input pattern is matched in a ‘fuzzy’ manner with stored patterns to form alignments where each alignment contains only two patterns: the input pattern and one of a set of stored patterns. This kind of matching can be achieved with any of the several varieties of dynamic programming or, indeed, the SP61 model itself. It is also the kind of matching which is performed by any of the many varieties of spelling checker.

It would be wrong to suppose that words should always be treated as simple patterns which are lacking in internal structure. Words are often constructed from more or less discrete substructures or ‘morphemes’ such as suffixes like ‘-ed’, ‘-ing’, ‘-tion’ or prefixes like ‘inter-’, ‘ultra-’, ‘un-’, and so on. However, for the purpose of illustrating PtR and IR which involves simple patterns, it is convenient to use (written) words as if they were lacking in internal structure apart from their composition from letters.

Thus, SP61 has been run using the thoroughly mis-spelled ‘word’, ‘`c m p u x t a r`’ as New and, as Old, a small dictionary of 45 words prepared by selecting, in a more or less haphazard manner, one, two or three words from each of the alphabetic sections of an ordinary English dictionary. Each word in the small dictionary was assigned a notional frequency of occurrence so that relative values corresponded roughly with their subjective rate of use in ordinary English.

Figure 1 shows the four alignments formed by SP61 which have the highest CDs. The first three are, by far, the best matches for the given pattern having CDs of (from the top alignment downwards) of 41.39, 29.14 and 28.44. The next best alignment is the fourth one shown in Figure 1, which has a CD of only 13.76. Notice that each of the first three alignments contain discrepancies between ‘`c m p u x t a r`’ and the word to which it has been aligned which represent all three of the possible kinds of discrepancy: omission, addition and substitution of symbols.

Figure 1 about here

Figure 1
The best four alignments formed by SP61 between '`c m p u x t a r`' and words in a small dictionary of 45 words. Compression scores for the four alignments are, from the top down, 41.39, 29.14, 28.44 and 13.76.

It is pertinent to point out that, in the small dictionary of words which was used as Old, the words '`c o m p u t e r`' and '`c o m m u t e r`' were assigned the same notional frequency of occurrence (10,000). Thus the relatively high CD of the first alignment in Figure 1 is not due to differences in frequency between the two words from Old. It is due to the fact that the best alignment encodes more symbols from New than any other alignment.

Regarding probabilities, the absolute probability of the best alignment is calculated by SP61 as 0.00652. Since there is no other alignment which contains all and only the symbols '`c`', '`m`', '`p`', '`u`', '`t`' and '`r`' from New, the reference list of alignments containing all and only the same symbols from New has only one entry, so the relative probability of the best alignment is 1.0.

In general, these results accord with our intuition that the 'correct' form of '`c m p u x t a r`' is '`c o m p u t e r`' and our intuition that '`c o m m u t e r`' is a very close alternative.

3.2.1 Inferences

What are the inferences when '`c m p u x t a r`' is recognised as '`c o m p u t e r`'? They are that between '`c`' and '`m`' there should be '`o`' and between '`t`' and '`r`' there should be '`e`'. The letters '`x`' and '`a`' in '`c m p u x t a r`' which do not form hits with any letter in '`c o m p u t e r`' are simply discarded, with no interpretation provided for them except, possibly, that they should not be present in that context.

Since these inference are derived from an alignment which is the sole member of its reference set, the probabilities of these inferfences are 1.0 like the alignment itself.

3.3 Patterns with internal structure: medical diagnosis

To provide an example with a rather different 'flavour', medical diagnosis has been chosen to show how ICMAUS may model the recognition or retrieval of patterns which do have internal structure, somewhat in the same way that sentences can be analysed into phrases, words and so on (see Section 3 of [2] and [5]).

It is pertinent to remark that, in Shortliffe's classic expert system (MYCIN [1]) and later systems of this kind, medical diagnosis has not normally been treated as a problem of PtR or IR. Rather, it has more commonly been treated as a problem requiring the application of 'rules' and the formation of 'chains' of inference. The example in this subsection, together with those in Section 4 (about the modelling by ICMAUS of networks, trees and rules) perhaps serve to show that, within the ICMAUS framework, the distinction between best-match PtR/IR and the more classical kinds of inference is not as sharp as may otherwise appear.

In Figure 2, the first pattern may be interpreted as an abstract description of an infectious disease with each of the first 17 numbers representing a category of symptom or sign but without any specific value for that category. Thus ‘`1`’ represents (the state of the patient’s) ‘`appetite`’, ‘`2`’ represents (the nature of the patient’s) ‘`breathing`’, and so on. Amongst the last 3 numbers, ‘`18`’ represents the organism which causes the disease, ‘19’ represents the mode of treatment of the disease, and ‘`20`’ acts as a terminating symbol for the whole pattern. Of course, these ‘numbers’ are merely symbols in the sense defined in Appendix A1 of [2] and they do not have any mathematical meaning.

```
disease #disease patient #patient R 1 2 3 4 5
    6 7 8 9 10 11 12 13 14 15 16 17 18 19 20          14755
disease chicken_pox #disease R fever 1 normal 2
    3 normal 4 no 5 no 6 no 7 8 no 9 10 normal
    11 yes 12 normal 13 normal 14 rash 15
    16 normal 17 no 18 chicken_pox_virus
    19 chicken_pox_treatment 20                        2500
disease influenza #disease R flu_symptoms
    1 normal 2 3 normal 4 6 no 7 8 no 9
    10 normal 11 no 12 14 normal 15 17 no
    18 flu_virus 19 flu_treatment 20                  10000
disease malaria #disease R fever 1 normal 2
    3 normal 4 yes 5 no 6 no 7 8 yes 9 yes
    10 normal 11 no 12 normal 13 normal
    14 sweating 15 16 normal 17 no
    18 malarial_protozoan
    19 anti_malaria_drugs 20                           2000
disease pneumonia #disease R fever 1 normal 2
    3 yes 4 no 5 yes 6 no 7 8 no 9 no 10 normal
    11 no 12 normal 13 normal 14 normal 15
    16 normal 17 no 18 pneumonia_bacterium
    19 pneumonia_treatment 20                           200
disease smallpox #disease R flu_symptoms
    1 normal 2 3 normal 4 6 no 7 8 no 9 10 normal
    11 no 12 14 rash_with_blisters 15
    17 no 18 smallpox_virus
    19 smallpox_treatment 20                              5
```

```
disease tuberculosis #disease 1 normal 2 short 3 pain
    4 no 5 yes 6 no 7 normal 8 no 9 no 10 normal
    11 yes 12 normal 13 normal 14 normal 15 normal
    16 normal 17 loss 18 Mycobacterium_tuberculosis
    19 tuberculosis_treatment 20                          50
R fever 1 2 rapid 3 7 flushed 8 15 high 16             14705
R flu_symptoms fever 1 4 yes 5 yes 6 9 yes 10
    12 aches 13 runny 14 16 sore 17                    10005
```

Figure 2
Six patterns describing six well-known infectious diseases, together with two patterns representing clusters of symptoms used in the descriptions of diseases. The figure on the right of each pattern is the notional frequency of occurrence of the corresponding disease or cluster of symptoms in an imaginary population in a given period. ***Key:*** `1 = appetite, 2 = breathing, 3 = condition of chest, 4 = chills, 5 = cough, 6 = diarrhoea, 7 = condition of face, 8 = fatigue, 9 = headache, 10 = lymph nodes, 11 = malaise, 12 = condition of muscles, 13 = condition of nose, 14 condition of skin, 15 = temperature, 16 = condition of throat, 17 weight changes, 18 causative agent, 19 = recommended treatment.`

Numbers are used instead of names because otherwise alignments formed by SP61 would be rather long and would, in many cases, be too long to fit comfortably on one page, even with the abbreviation of the alphabetic symbols which will be seen below. Notice that this is purely a matter of presentation and has nothing to do with the information ‘cost’ (in bits) which is calculated for each symbol or the compression performed by the program.

As we shall see, the first pattern provides a ‘framework’ for other patterns and symbols so that alignments can be formed which can unify without mis-matches (see Section 4.8.4 of [2]).

Amongst the remaining patterns in Figure 2, the first six describe the attributes of six well-known infectious diseases. The last two patterns describe two named ‘clusters’ of symptoms (‘fever’ and ‘flu symptoms’).[4] Each of the two names are used within the disease patterns as a shorthand for the corresponding cluster of symptoms.

Notice that each number representing a category is placed to the left of the value for that category and, in addition, the following number is placed to the right of the value even if there is no value for the following category. Using the following number as a ‘termination symbol’ like this is necessary to avoid the ambiguity of symbols like ‘`yes`’ and ‘`normal`’ each of which can apply to several different categories. The ‘slot’ between one number and the following number functions like a ‘variable’ in conventional systems (see also Section 4.1).

---

4. These eight patterns are approximations to the information provided in *The British Medical Association Complete Family Health Encyclopaedia*, London: Dorling Kindersley, 1990.

To the right of each pattern is a number representing the notional frequency of occurrence of the pattern in an imaginary population in a given period. The frequency for the first pattern may be interpreted as the global frequency of infectious diseases of the kinds represented in the knowledge base. The frequency of each of the next six patterns is the frequency of occurrence of the corresponding disease.[5] The frequency of each of the last two patterns is the notional frequency of the corresponding cluster of symptoms.

3.3.1 Alignments

Figure 3 shows the two best alignments formed by SP61 with the pattern '`patient John #patient 1 poor 2 rapid 3 4 yes 5 yes 6 7 flushed 8 yes 9 10 normal 11 no 12 aches 13 runny 14 15 high 16 sore 17`' as New and the patterns from Figure 2 as Old. Notice that some categories (`3`, `6`, `9`, `14`, `17`, `18` and `19`) have no 'value'. This should be interpreted as meaning that no relevant information had been obtained for these categories at the time the pattern was compiled. It does *not* mean that the relevant values were normal or negative.

In the alignments shown in Figure 3, many of the alphabetic symbols have been abbreviated in accordance with the key shown with the figure. As with the use of number symbols for categories, these abbreviations are used so that alignments can be shown within the confines of one page and they have nothing to do with the information cost of the symbols or the compression performed by SP61.

The pattern for influenza appears in row 4 of Figure 3 (a) while the pattern for smallpox appears in row 4 of Figure 3 (b). Thus Figure 3 (a) may be interpreted as a diagnosis of influenza for the patient 'John' while Figure 3 (b) may be construed as a diagnosis of smallpox for the same patient.

3.3.2 'Clusters' and 'references' within alignments

Both influenza and smallpox are characterised by 'flu symptoms' described by the pattern which appears in row 3 of both alignments. However, within the flu symptoms cluster, there is a subsidiary cluster called 'fever' described by the pattern which appears in row 2 of each alignment. In the 8th column of each alignment, there is a link which may be interpreted as a 'reference' from the disease pattern to the pattern for flu symptoms. And in the 9th column of each alignment, there is a link which may be interpreted as a 'reference' from the pattern for flu symptoms to the pattern for fever.

---

5. For the purpose of this exercise, we shall suppose, contrary to current fact, that cases of smallpox do still occasionally appear.

Figure 3 about here

Figure 3

The two best alignments formed by SP61 with the pattern 'patient John #patient 1 poor 2 rapid 3 4 yes 5 yes 6 7 flushed 8 yes 9 10 normal 11 no 12 aches 13 runny 14 15 high 16 sore 17' as New and the patterns shown in Figure 2 as Old. ***Key:*** as = aches, bs = blisters, fd = flushed, fls = flu_virus, fr = fever, fs = flu_symptoms, ft = flu_treatment, hh = high, ia = influenza, Jn = John, nl = normal, pr = poor, pt = patient, rd = rapid, rs = rash_with_blisters, ry = runny, se = sore, smt = smallpox_treatment, ss = smallpox_virus, sx = smallpox, ys = yes.

3.3.3 Compression scores and probabilities

Using the methods of calculation described in Section 5 of [2] and Section 2 of this article, measures of compression and probability for Figure 3 (a) and (b) are calculated as shown in Table 5.

Alignment	B_N	B_E	CD	p_{ABS}	p_{REL}
Figure 3 (a)	294.42	13.40	281.02	9.2482e-5	0.9980
Figure 3 (b)	294.42	22.35	276.21	1.8768e-7	0.0020

Table 5

Measures of compression and probability for Figure 3 (a) and (b). ***Key:*** B_N is the number of bits in New. B_E is the calculated cost, in bits, which is required to encode New in terms of the patterns in Old which appear in the given alignment. CD is the 'compression difference' (CD = B_N - B_E). p_{ABS} is the 'absolute probability' of each alignment. And p_{REL} is the 'relative probability' of each alignment.

These two alignments constitute a 'reference set' of alignments because they encode the same symbols from New and there are no other alignments which encode these symbols. Thus, their relative probabilities may be calculated as shown in Table 5. These probabilities accord with our intuition that, given the appearance of flu symptoms and without any other information about symptoms, a diagnosis of influenza is, in recent years, very much more likely than a diagnosis of smallpox.

This large discrepancy between the assessed probability of influenza and the probability of smallpox has been obtained despite the fact that the pattern for New does not contain the critical feature which distinguishes smallpox from influenza: a skin rash together with blistering of the skin. Of course, any doctor who examined the patient in the normal way would see immediately whether or not the patient had the rash and blistering of the skin which is so characteristic of smallpox. Let us suppose, for the sake of our discussion, that this examination of the patient was made via the telephone!

What happens if information about the condition of the patient's skin has been obtained and that this is recorded in New (by inclusion of the symbol '`rash_with_blisters`' after the symptom category '`14`')? In this case, the two best alignments in terms of CD have the same structure as before except that, in the case of smallpox, there is a hit between the symbol '`rash_with_blisters`' in New and the matching symbol in Old. Relevant measures for the two alignments are shown in Table 5.

Alignment	B_N	B_E	CD	p_{ABS}
'Influenza'	292.83	13.40	279.43	9.2482e-5
'Smallpox'	331.38	21.75	309.63	2.8337e-7

Table 6

Measures of compression and probability for the best two alignments obtained by SP61 when Old is the patterns shown in Figure 2 and New is the same as in Figure 3 except that it includes '`rash_with_blisters`' between '`14`' and '`15`'. ***Key:*** Column headings are the same as in Table 5. 'Influenza' and 'Smallpox' mean the alignments containing the patterns for influenza and smallpox respectively.[6]

As in Table 5, the absolute probability of the 'influenza' alignment is much higher than for the 'smallpox' alignment. However, by contrast with the results shown in Table 5, the alignment for smallpox has the highest CD. This is because it encodes one more symbol from New than the alignment for influenza.

As was indicated in Section 2, absolute probability is not, in itself, a good means of comparing one alignment with another or for calculating relative probabilities. This is because it confounds two things: the amount of information from New which can be encoded in terms of Old and the effectiveness of the encoding itself. To evaluate alignments in relation to each other we need to compare alignments which encode the same symbols from New.

In this case, the 'smallpox' alignment encodes more symbols from New than any other and so it belongs in a reference set of alignments by itself. This means that, although the absolute probability of this alignment is very low (2.8337e-7), the *relative* probability of the smallpox alignment, in relation to the reference set of symbols in New, is 1.0! There is no value for the relative probability of the 'influenza' alignment because it is not a member of the reference set of alignments.

In a similar way, if the condition of the skin is marked explicitly as '`normal`', then the relative probability of the 'influenza' alignment in relation to other alignments which encode the same symbols from New will be 1.0.

In accordance with the remarks in Section 2.9, a calculated probability of 1.0 does not mean that we should have total confidence in the apparent diagnoses - if only because the knowledge base in this case is so far from being a comprehensive description of infectious human diseases.

---

6. Readers may wonder why, in Table 6, the value for B_N for the 'influenza' alignment is lower than the corresponding value in Table 5 despite the fact that the same symbols from New are encoded in each case. The reason is that the gap between the symbols '`14`' and '`15`' is larger in the case of Table 5 because of the interpolation of the symbol '`rash_with_blisters`'.

3.3.4 Inferences

In accordance with what was said in Section 2, the inferences associated with any alignment are the symbols, substrings and subsequences of patterns from Old which appear in the alignment and which are not matched to any symbols in New.

With regard to Figure 3 (a), we may infer that the patient has influenza and that, if we had more information about the patient, we would find that the condition of his or her chest (category '`3`') would be normal, that the patient would not have diarrhoea (category '`6`'), and so on. In particular, we would infer that the patient's skin would *not* have the combination of rash and blisters which is characteristic of smallpox (category '`14`').

In terms of practicalities, there are two important inferences that we can make from an alignment like that in Figure 3 (a):

- We can infer that the influenza virus will be present in the patient's blood. This may assist us in gathering further information about the patient's condition or in determining the best treatment.
- If we are satisfied that a given alignment represents a sound diagnosis, we can infer from the alignment how the disease should be treated (category '`19`'). The recommended treatment for a given disease is simply a part of the pattern which the process of diagnosis has identified.

3.3.5 Mis-matches

What about a category like '`1`' ('appetite') where, in Figure 3 (a), there is a mis-match between what was observed in the patient (poor appetite) and the expectation from the pattern for influenza that the patient's appetite would be normal? Likewise for category '`8`'?

There is, of course, a host of reasons why the diagnostic information might not coincide exactly with a text-book definition of a disease: the diagnosis may be wrong and the patient has some disease which is new to science or, at least, not recorded in the knowledge base, the patient or other person giving information to the doctor may have some reason to be misleading, the food which is available to the patient may be dull, there may have been a clerical error, and so on.

3.3.6 Inference as a request for more information and refinement of a diagnosis

Apart from the recommended treatment, any of the inferences which we can read from an alignment may be regarded as a request for more information about the patient. For example, the inference that the 'flu virus is likely to be found in the patient's blood may be regarded as a request for the appropriate test to be made.

When additional information has been obtained about the patient, it may be added to New and the program may be run again. The new results may strengthen the previous diagnosis or point to one or more alternative possible diagnoses. Inferences drawn from the newly-formed alignments may lead to the gathering of yet more information about the patient and the formation of further alignments and inferences. By iteration in this way, the diagnosis may be progressively refined.

3.3.7 Diagnosis of two or more diseases in a patient at one time

The discussion so far and the examples which have been presented may suggest that diagnosis means identification of a single disease which best accounts for the patient's symptoms and signs. However, it is possible for any one patient to have two or more diseases at one time: a patient may have combinations of diseases like influenza with bronchitis, common cold with athlete's foot, and so on.

These kinds of possibility sit comfortably within the ICMAUS framework. It is entirely possible for an alignment to contain patterns for two or more different diseases. However, any such alignment will only occur amongst the 'best' alignments for a given set of symptoms and signs if the encoding 'cost' of the additional pattern or patterns is offset by the encoding of more symbols from New than would be the case if the alignment contained a pattern for only one disease.

3.4 Class hierarchies with inheritance

For many years, it has been recognised that objects and patterns can often be grouped into 'classes' where members of a class are similar in the sense that they have some attributes in common (but see Section 3.5, next). It has also been recognised that classes may themselves be grouped into classes recursively through any number of 'levels'. Thus two or more chairs which are not identical may belong in the class 'chairs'; 'chairs' and 'tables' may be assigned to the class 'furniture', and so on. This kind of class hierarchy has been given a formal status by biologists in the classification of animals and plants.

Since the mid-sixties, when Simula was invented, it has been recognised that, with an appropriate 'object-oriented' language like Simula or C++, software may be designed to model real-world 'objects' and 'classes' in such a way that the 'attributes' of a given class (functions or data structures) may be represented just once within the class and then 'inherited' through any number of levels by the objects and classes within the given class. Amongst the several advantages of this style of software design is the relative conciseness or compression of software because structures and functions do not need to be replicated in different parts of a system.

Inheritance is a form of inference which is similar to the 'probabilistic' kind of 'modus ponens' reasoning which will be discussed in Section 4.1. Knowing that 'Tweety' is a member of the class 'bird' tells us that Tweety (probably) has the attributes of a bird. Inheritance allows us to go further than this kind of one-step reasoning. We can follow the inferences up through the class hierarchy from 'birds' to 'vertebrates', from 'vertebrates' to 'animals', from animals to 'living thing', and even further to 'physical object' and perhaps beyond. At each stage, there are new things which we can infer about the humble Tweety.

3.4.1 Modelling class hierarchies and inheritance with patterns

Figure 4 shows some patterns representing the class 'vertebrates' with some of its subclasses. The symbols in each pattern are indented so that the internal structure of each pattern can be easily seen. The first pattern, representing the class 'vertebrates' contains a 'description' which itself contains a 'name', 'structure', 'voice', 'respiration', 'reproduction' and 'habitat'. 'structure' contains 'head', 'body', 'legs', and so on.

Each substructure has an initial symbol which names the structure and a matching 'termination' symbol which begins with the hash character ('#'), in much the same way that the beginnings and ends of 'nouns', 'verbs' and 'sentences' were marked in the parsing example in Section 3 of [2].

In Figure 4, the second and subsequent patterns represent subclasses of 'vertebrates' at the levels of child ('reptile', 'bird', 'mammal'), grandchild ('marsupial', 'cetacean', 'carnivore') and great-grandchild ('cat', 'dog').

As in object-oriented computer languages, each class in Figure 4 is linked to its parent class (if any) by including the relevant symbol or symbols for the parent class in the definition of the child class. The definition of 'cat' includes the fact that it is a carnivore, the definition of carnivore includes the fact that it is a mammal, and so on. In our example, the class 'vertebrate' has no parent class although in a more fully-developed example it may be shown as part of the class 'animal'.

3.4.2 Multiple alignment, inference and inheritance

Figure 5 shows the best alignment found by SP61 between the pattern '`name Tibby #name backbone hair blood warm #blood purrs nursing milk #nursing land`' in New with the patterns from Figure 4 in Old. As before, the length of the alignment has made it necessary to abbreviate the symbols and to split the alignment into sections. As before, row numbers are provided at the beginning and end of each section to facilitate reading of the alignment.

In accordance with the convention which has been adopted, the top row contains the pattern from New while the other rows show patterns from Old in an arbitrary order. Row 4 identifies the unknown animal as a cat which is a carnivore (row 3) and a mammal (row 1) and a vertebrate (row 2). Each row encodes one or more of the symbols from New.

```
vertebrate
     description
          name #name
          structure
               head
                    cranium
                    mouth_parts #mouth_parts
               #head
               body
                    backbone
               #body
               legs
                    number #number
                    feet #feet
               #legs
               form #form
               covering #covering
```

```
                blood #blood
            #structure
            voice #voice
            respiration #respiration
            reproduction #reproduction
            habitat #habitat
        #description
#vertebrate                                         12000

vertebrate
        reptile
            description
                structure
                    covering
                        dry_skin
                    #covering
                    blood cold #blood
                #structure
                respiration lungs #respiration
                reproduction
                    eggs_in_shells_on_land
                #reproduction
            #description
        #reptile
#vertebrate                                         5000
vertebrate
        bird
            description
                structure
                    mouth_parts
                        beak
                    #mouth_parts
                    covering feathers #covering
                    blood warm #blood
                #structure
                reproduction
                    eggs_in_shells_on_land
                #reproduction
            #description
        #bird
#vertebrate                                         3000

vertebrate
        mammal
            description
                covering hair #covering
```

```
                blood warm #blood
                reproduction
                     birth live #birth
                     nursing milk #nursing
                     size_at_birth #size_at_birth
                     protection #protection
                #reproduction
           #description
      #mammal
#vertebrate                                                        8000

mammal
      marsupial
           description
                reproduction
                     size_at_birth
                          small
                     #size_at_birth
                     protection pouch #protection
                #reproduction
           #description
      #marsupial
#mammal                                                            1000
mammal
      cetacean
           description
                form fish_like #form
                habitat sea #habitat
           #description
      #cetacean
#mammal                                                            2000

mammal
      carnivore
           description
                legs
                     number four #number
                #legs
                habitat land #habitat
           #description
      #carnivore
#mammal                                                            2500

carnivore
      cat
           description
```

```
                feet retractile_claws #feet
                voice purrs #voice
           #description
      #cat
#carnivore                                             500

carnivore
      dog
           description
                voice barks #voice
           #description
      #dog
#carnivore                                             700
```

Figure 4
Patterns representing the class 'vertebrates' with some of its subclasses. The number associated with each pattern is a notional frequency of occurrence.

Figure 5 about here

Figure 5
The best alignment found by SP61, between the pattern '`name Tibby #name backbone hair blood warm #blood purrs nursing milk #nursing land`' in New and the patterns in Figure 4 in Old. Owing to its length, the symbols are abbreviated according to the key, below, and the alignment is split into two sections, with row numbers at the beginning and end of each section to facilitate the reading of the alignment. ***Key:*** `Ty = Tibby, be = backbone, bh = birth, bld = blood, by = body, ce = carnivore, ct = cat, cg = covering, cm = cranium, dn = description, ft = feet, fm = form, fr = four, ht = habitat, hr = hair, hd = head, ld = land, ls = legs, le = live, ml = mammal, mk = milk, ms = mouth_parts, ne = name, nr = number, ng = nursing, pn = protection, ps = purrs, rn = reproduction, rsn = respiration, rs = retractile_claws, sh = size_at_birth, se = structure, ve = vertebrate, vce = voice, wm = warm.`

As with other examples in this article, the inferences expressed by the alignment in Figure 5 are represented by the symbols in the alignment, in any row below the first row, which are not matched to any symbol in New.

In this example, *inference is inheritance*: we say that the animal 'inherits' the attributes of all the classes in which it belongs. Identifying the unknown animal as a cat allows us to infer that it has retractile claws ('rs'). Knowing that it is a carnivore tells us that it has four legs ('`ls nr fr #nr ... #ls`'). Membership of the class mammal means that the animal's young are born 'live' ('`bh le #bh`'), i.e., not in eggs. And belonging in the class 'vertebrate' means that the animal has a head ('`hd`') with a cranium ('`cm`') and mouth parts ('`ms`'), a body, legs, some kind of form (shape), some kind of covering, blood, a voice, some means of respiration, and so on.

3.4.3 Multiple inheritance

'Natural' objects and classes that we recognise in everyday speaking and thinking often have more than one parent class: one person may be both a 'woman' and a 'doctor', another person may be both a 'man' and a 'footballer', and so on. In this kind of 'multiple inheritance', the parent classes overlap without one being a subclass of the other.

It is not hard to see that the ICMAUS model can accommodate this kind of relationship between classes. In much the same way that MA can model the fact that a patient can have more than one disease at one time, multiple inheritance can be modelled by the appearance in one alignment of two or more patterns representing classes which are aligned with New but not with each other, unless they happen to have some attributes in common.

3.5 Polythetic classes

'Natural' classes are more subtle than the foregoing discussion may suggest. A curious feature of the classes we use in everyday speaking and thinking is that no single attribute need be present in every member of the class. This kind of 'polythetic' class may be defined by attributes each one of which is normally present in every member of the class but may be absent here and there. In much the same way that we can recognise a pattern or object, when it is partially obscured, by almost any (reasonably large) subset of its parts (Section 7.1 of [2]), an entity can be recognised as belonging in a given class provided it exhibits any reasonably large subset of the attributes of the class.

In the tiny fragment of English grammar discussed in Section 3.1 of [2], the set of possible sentences defined by the grammar is a polythetic class as just described. Each one of the patterns '`John runs`', '`Susan runs`', '`John walks`', '`Susan walks`' belongs in the class 'sentence' but none of the 4 words in the grammar (which we may regard as 'attributes' of the sentence) are present in every instance.

Thus polythesis is not as mysterious as it may at first appear. As we saw in Section 3.1 of [2], it can be modelled with a simple phrase-structure grammar and it can also be modelled in the ICMAUS framework. More sophisticated examples in the realm of syntax and parsing are presented in [5].

It is not hard to see that the same mechanisms and principles can be transferred directly to patterns which represent non-linguistic objects and classes.

4 Reasoning with networks, trees and rules

This section presents examples of reasoning which are a little more elaborate than the example presented in Section 7.2 of [2]. Examples include reasoning in one step (Sections 4.1 and 4.2) and 'chains' of reasoning in one or more stages using a network or tree (Section

4.3) or rules in an expert system (Section 4.4).

Bayesian networks are not described or discussed in this section. However, Section 6 of [4] (the third of these three articles) shows how ICMAUS may be provide an alternative to Bayesian networks as a means of explaining the phenomenon of 'explaining away'.

4.1 'Deductive' reasoning

Consider a 'standard' example of *modus ponens* syllogistic reasoning:

1 $\forall x$: bird(x) $\supset$ canfly(x)
2 bird(Tweety)
3 $\therefore$ canfly(Tweety).

which, in English, may be interpreted as:

1 If something is a bird then that something can fly.
2 Tweety is a bird.
3 Therefore, Tweety can fly.

In strict logic, a 'material implication' like ($p \supset q$) ("If something is a bird then that something can fly") is equivalent to ($\sim q \supset \sim p$) ("If something cannot fly then it is not a bird") and also to ($\sim p \vee q$) ("Either something is not a bird or it can fly"). However, there is a more relaxed, 'everyday' kind of 'deduction' which, in terms of our example, may be expressed as: "If something is a bird then, *probably*, it can fly. Tweety is a bird. Therefore, *probably*, Tweety can fly."

This kind of probabilistic 'deduction' differs from strict material implication because it does not have the same equivalencies as the strict form. If our focus of interest is in describing and reasoning about the real world (rather than exploring the properties of abstract systems of symbols), the probabilistic kind of 'deduction' seems to be more appropriate. With regard to birds, we know that there are flightless birds, and for most other examples of a similar kind, an "all or nothing" logical description would not be an accurate reflection of the facts.

A simple and natural way of recording something like the association between being a bird and being able to fly is to place a symbol corresponding to a bird next to or near to a symbol corresponding to the ability to fly, something like this: '`bird canfly`'. In other words, an association can be expressed as a *pattern*, as defined in Appendix A1 of [2]. In this context, an 'association' is two or more entities or events which occur together in the world with a relatively high frequency.[7,8]

Using patterns, how should we record the other attributes of a bird? The simplest and most natural way is in a single pattern like this:

---

7. At this point, readers may object that symbols like '`bird`' `and` '`canfly`' have a meaning which is not visible, thus breaking the principle of "no hidden meaning" (Appendix A1 of [2]). However, as specified in the definition of *symbol* in that appendix, the principle applies only to the behaviour of a symbol in the system, not the meaning which the symbol may have for a human reader.

```
bird name #name canfly wings feathers beak
     crop lays_eggs ... #bird
```

which records all the attributes which are, at one and the same time, associated with the concept '`bird`' and with each other.[9,10] A pattern of this kind captures the main features of a bird and would provide an overall framework for the structure and functions of birds in any computer-based modelling system.

How may such a pattern enter into MA and PrbRs? We may suppose that this pattern and others of a similar kind are stored in 'Old' and that a pattern, '`bird Tweety`', corresponding to the proposition "Tweety is a bird" or 'bird(Tweety)' is supplied as New information to be matched with Old information.

In the main pattern, the pair of symbols '`name #name`' functions as a 'variable' (as will be discussed in Section 1.1 of [4]). It is a location in the knowledge structure which may receive any one of a range of different 'values'. Like a variable in any conventional system, it may have a 'type' definition which specifies the range of different values which may appear in the variable. In the present scheme, these potential values of the variable can be defined extensionally as a set of patterns like '`name John #name`', '`name Susan #name`', '`name Tweety #name`' and so on. These supplementary patterns should also be stored in Old.

Given patterns like these in New and Old, as shown in Figure 6, the best alignment found by SP61 is the one shown in Figure 7.

---

8. As discussed in [2], "relatively high frequency" means a frequency of occurrence in a reference 'corpus' of data which, for a given pattern, is higher than one would expect for patterns of the same size if the corpus were random. The frequency of occurrence can include occurrences of the pattern as either or both of substrings or subsequences within the corpus.

This concept (but restricted to substrings) has been the foundation of a system (MK10) for the unsupervised learning of word patterns in unsegmented natural language text and a system (SNPR) for the unsupervised learning of artificial grammars from unsegmented text (see [6, 7], and earlier papers referenced therein).

9. Since the concept of 'pattern' includes arrays of two or more dimensions, the structural attributes may, in some future version of the SP system, be represented in a diagram or picture, with or without labels for each part. As previously noted, patterns in two or more dimensions are outside the scope of the SP model as it has been developed so far.

```
bird name #name canfly wings feathers
     beak crop lays_eggs ... #bird                   1000
fish name #name gills fins scales tail ... #fish   1200
bat name #name fur canfly eats_insects ... #bat     500
tortoise name #name shell scaly_skin ... #tortoise  800
name Susan #name                                    900
name Peter #name                                    600
name Tweety #name                                   700
name Tibby #name                                    400
```

Figure 6

A small knowledge base showing some of the attributes of four different animals and some names. As in previous examples, the number to the right of each pattern is an imaginary frequency of occurrence in some notional sample of the world.

Figure 7 about here

Figure 7

The best alignment found by SP61 with the pattern '`bird Tweety`' in New and the patterns shown in Figure 6 in Old.

As before, the inferences which are expressed by this alignment are represented by the unmatched symbols in the alignment. The fact that Tweety is a bird allows us to infer that Tweety can fly but it also allows us to infer that Tweety has wings, feathers and all the other attributes of birds. These inferences arise directly from the pattern describing the attributes of birds.

There is only one alignment which encodes all the symbols in New. Therefore, the relative probability of the alignment is 1.0, the relative probability of '`canfly`' is 1.0, and likewise for all the other symbols in the alignment, both those which are matched to New and those which are not.

At this point readers may wonder whether the ICMAUS scheme can handle non-monotonic reasoning: the fact that additional information about penguins, kiwis and other flightless birds would invalidate the inference that something being a bird means that it can

---

10. The symbol '`#bird`' functions as a 'termination symbol' which is not strictly necessary in this case but is included because it would be required if a 'reference' to the given pattern were embedded in any larger pattern. As will be seen shortly, the '`#name`' symbol also functions as a 'termination symbol' which is needed for the formation of an alignment which can unify without 'mis-matches'. The expression 'termination symbol' has no formal status. As explained in Appendix A1 of [2], all symbols in the system (with one exception in SP61, introduced as a stop-gap measure to solve a problem in the scoring of alignments) are simply 'marks' which can be discriminated in a yes/no manner from each other and which enter into matching and unification in the same way.

fly. Discussion of this point is deferred until [4].

4.2 Abductive reasoning and the integration of abductive reasoning with probabilistic 'deductive' reasoning

In the ICMAUS framework, any subsequence of a pattern may function as what is 'given' in reasoning, with the complementary subsequence functioning as the inference. Thus, it is just as easy to reason in a 'backwards', abductive manner as it is to reason in a 'forwards', deductive manner. We can also reason from the middle of a pattern outwards, from the ends of a pattern to the middle, and many other possibilities. In short, the ICMAUS framework allows seemless integration of probabilistic 'deductive' reasoning with abductive reasoning and other kinds of reasoning which are not commonly recognised.

Figure 8 shows the best alignment and the other member of its reference set of alignments which are formed by SP61 with the patterns shown in Figure 6 in Old and '`Susan canfly`' in New.[11]

Figure 8 about here

Figure 8
The best alignment and the other member of its reference set of alignments which are formed by SP61 with the patterns shown in Figure 6 in Old and '`Susan canfly`' in New.

By contrast with the previous example where New contained '`bird Tweety`', there are two alignments in the reference set of alignments that encode all the symbols in New. These two alignments represent two alternative sets of abductive inferences that may be drawn from this combination of New and Old:

- With regard to the first alignment, Susan could be a bird with all the attributes of birds including the ability to fly. The relative probability of the alignment is 0.8, as is the relative probability of the pattern for '`bird`' and every other symbol in that pattern (apart from the '`name`' and '`#name`' symbols where the relative probability is 1.0).
- Alternatively, we may infer from the second alignment that Susan could be a bat. But in this case the relative probability of the alignment, the pattern for '`bat`' and all the symbols in that pattern (apart from the '`name #name`' symbols) is only 0.2.

---

11. Readers will notice that, to obtain these two alignments, it is necessary to put the name first instead of last as in the previous example. For the order of symbols to be constrained like this seems a little artificial but it is no different from established systems for reasoning like Prolog, where the order of symbols is critically important. Even in natural languages, the order of words cannot be changed unless suitable markers are included to assist interpretation (like those, in English, that distinguish active and passive forms).

4.3 Decision networks and decision trees

One of the simplest kinds of system which, in addition to reasoning in one step like the preceding examples, supports reasoning in two or more steps, is a ‘decision network’ or a ‘decision tree’. In such a system, a path is traced through the network or tree from a start node to two or more alternative destination nodes depending on the answers to multiple-choice questions at intermediate nodes. Any such network or tree may be given a probabilistic dimension by attaching a value for probability or frequency to each of the alternative answers to questions at the intermediate nodes.

Figure 9 shows a set of patterns, each of which represents a non-terminal node of a decision network from a car maintenance manual for diagnosing faults in a car engine. To save space, the text associated with each pattern has been omitted. Terminal nodes have also been omitted because, without the text, each one would contain nothing but a number symbol. Figure 10 shows a sample of the patterns for non-terminal and terminal nodes of the network with the text for each node included.

In Figure 9, each pattern except the pattern for the start node is identified by the number symbol which appears at the beginning of the pattern, together with the ‘`yes`’ or ‘`no`’ answer to the question in the parent node. The number at the end of each pattern identifies the two children of the node.

As usual with an ICMAUS knowledge base, each pattern in Figure 9 has a frequency of occurrence, shown to the right of each pattern. In the present case, each frequency is a fictional frequency of a symptom or symptoms in the notional domain of a car mechanic or group of car mechanics. These figures have been invented by the author. They were not present in the original fault-finding chart on which this example is modelled.

```
Start 43                                             1202
    43 yes 44                                        1043
        44 yes 19                                    1009
            19 yes 59                                  46
            19 no 1                                   963
                1 yes 2                               691
                    2 yes 4                            92
                        4 yes 58                       36
                        4 no 23                        56
                    2 no 5                            599
                        5 yes 58                       62
                        5 no 27                       537
                            27 yes 21                 293
                                21 yes 61              84
                                21 no 22              209
                                    22 yes 24          24
                                    22 no 22a          87
                                        22a yes 35     44
                                        35 yes 58      19
                                            35 no 26   25
```

```
                                        22a no 36     43
                                            36 yes 58 21
                                            36 no 26  22
                            27 no 60                 244
                1 no 3                               272
                    3 yes 6                           36
                    3 no 7                           236
                        7 yes 8                       71
                            8 yes 8a                  55
                            8 no 9                    16
                        7 no 10                      165
                            10 yes 11                115
                                11 yes 13             14
                                11 no 14             101
                                    14 yes 15         21
                                    14 no 16          80
                            10 no 12                  50
        44 no 51                                      34
            51 yes 56                                 14
            51 no 59                                  20
    43 no 45                                         159
        45 yes 46                                    145
            46 yes 52                                120
                52 yes 53                             86
                52 no 54                              34
                    54 yes 55                         22
                    54 no 50                          12
            46 no 62                                  25
        45 no 47                                      14
            47 yes 48                                  8
            47 no 49                                   6
```

Figure 9

A set of patterns representing the non-terminal nodes of a decision tree for the diagnosis of faults in a car engine.[12] To save space, the text associated with each pattern has been omitted. The full version of some of these patterns can be seen in Figure 10 together with two of the patterns for terminal nodes.

12. From *Car Care, How It Works, How to Look After It*, London: Publication Division of the Automobile Association, 1983.

```
Start 43 Does the starter turn the engine?
    43 yes 44 Does starter turn the engine briskly?
        44 yes 19 Slowly press throttle to floor, return
                choke and try again.Does engine start?
            19 yes 58
            19 no 1 Is there a spark at the plug leads?
                1 yes 2 Remove and examine plugs.
                        Are they black, oily or wet?
                    2 yes 4 Thoroughly clean and
                        dry [plugs], or renew, check
                            gap and refit.Try again.
                                Does engine start?
                        4 yes 58
                        4 no 23
                    2 no 5 Clean, check gap and replace
                        [plugs].Does engine start?
(etc)
23 Check for water in fuel. Drain off to clear.
58 Problem is solved.
(etc)
```

Figure 10

A sample of the patterns representing nodes of the decision network for diagnosing faults in car engines, including the text associated with each pattern. Patterns for non-terminal nodes are shown at the top. The two patterns at the bottom represent terminal nodes.

4.3.1 Networks and trees: convergence and divergence of links

The patterns in the full network of non-terminal and terminal nodes represent a network rather than a tree because there is both divergence and convergence of links. Divergence appears when a node has two or more children while convergence appears where a node has two or more parents (e.g., node 58). Clearly, a decision tree can be represented by using simple patterns in the same manner as this example of a decision network, except that there must not be any patterns which have two or more parents.

4.3.2 Forming multiple alignments: sequential processing of New

How can this network be used to diagnose a fault in a car engine? In principle, this means putting the '`Start`' symbol into New followed by a sequence of '`yes`' and '`no`' symbols, putting the patterns from Figure 9 into Old, and then searching for the alignment which represents the best encoding of New in terms of Old.

This abstract description of how the system may be used is not very practical - because it leads to inefficient searching (in terms both of speed and success in finding the 'correct' alignment) and because it does not allow the questions which are posed by the system to be answered in the kind of progressive interactive way which is natural for this kind of system.

For these reasons, SP61 has been designed (as shown in Figure 6 of [2]) so that, where necessary or appropriate, it can be set to search for a good alignment by processing any pattern in New in sections or 'windows', one window at a time, in left-to-right order, forming intermediate alignments at each stage. The size of the window can vary from one symbol to the whole of New.

In this mode of operation, with each window containing only one symbol, it is possible to start the search with only the '`Start`' symbol in New and then add '`yes`' and '`no`' symbols to it one at a time so that one symbol is added whenever the system has finished processing the preceding symbols.

4.3.3 Multiple alignment and diagnosis

Figure 11 shows the best alignment found by SP61 with the patterns from Figure 9 in Old and the pattern '`Start yes yes no no no yes no`', supplied one symbol at a time to New.

Figure 11 about here

Figure 11

The best alignment found by SP61 with the pattern '`Start yes yes no no no yes no`' in New and the patterns from Figure 9 in Old. SP61 was set to operate in the progressive manner described in the text.

In this alignment, every symbol in a pattern from Old which does not form a hit with a symbol in New represents an inference which can be drawn from the alignment. However, the most important of these inferences is the last one, corresponding to the symbol '`9`' which represents a terminal node containing the text "There is an HT fault or, possibly, a capacitor fault". This is the diagnosis which the system offers in response to the sequence of yes/no symbols in New representing answers to questions posed by the system at intermediate stages. All the other 'inferences' - the symbols '`43`', '`44`', '`19`' etc - merely represent the questions which have been answered by the yes/no symbols in New.

4.3.4 Probabilities

What about probabilities? In this case, the reference set of alignments (which encode all and only the same symbols from New as the best alignment) contains only one member - the best alignment itself. In this case, the relative probability of the diagnosis is 1.0. In other words, this probabilistic version of the system provides an all-or-nothing answer in the same manner as the non-probabilistic chart from which it was derived. Of course, this system, like any other, is still subject to the law of "rubbish in means rubbish out". The confidence which we place in the conclusions must be qualified in the several ways that were noted in Section 2.9.

4.3.5 So what?

Regarding the example which has been discussed, readers may object that ICMAUS is a long-winded way to achieve something which is done perfectly adequately with a conventional expert system or even the original chart on paper from which the example was derived. Has anything been gained by re-casting the example in an unfamiliar form?

In this case, the answer is "probably not". The main reason for including the example in this article is to show that multiple alignment as it has been developed in the ICMAUS framework has a much broader scope than may, at first sight, be assumed.

For any particular domain, a system with this kind of broad scope may not have any particular advantage over a system which is dedicated to that domain and can function effectively only in that domain. However, a generic system which can embrace a range of more specific types of system can have substantial advantages, as described in Section 1.3.1 of [2].

4.3.6 Information which is incomplete

Another possible response to the "So what?" question, above, is that the SP system, unlike most conventional systems, does not depend exclusively on input which is both complete and accurate. One of the strengths of the ICMAUS framework is that it can bridge gaps in information supplied to it (as New) and can compensate for symbols which have been added or substituted in the input, provided there are not too many.

It is not immediately obvious why this kind of capability might be useful for a diagnostic system like the example above but it is interesting to see that SP61 can produce a plausible result even when parts of the input (in New) are missing or when there is addition or substitution of 'wrong' symbols.

SP61 has been run in sequential processing mode (so that each window contains only one symbol) with the patterns from Figure 9 in Old and the pattern '`Start 43 yes 44 7 yes no`' in New.

Apart from the addition of some 'correct' number symbols (for a reason which is given in a moment), this pattern in New is the same as in Figure 11 except that the middle sequence of yes/no symbols ('`yes no no no`') is missing. By contrast with the example in Figure 11, it is necessary in this case to include in New some of the number symbols from the relevant patterns in Old, otherwise there is too much ambiguity about which questions are being answered by the symbols '`yes`' and '`no`'.[13]

Figure 12 shows the best alignment produced by SP61 with Old and New as just described. No other alignment is produced which encodes all the symbols in New. The alignment in Figure 12 successfully bridges four steps in the alignment where information from New is missing and arrives at the same diagnostic conclusion as the alignment in Figure 11. By bridging this gap in New, the alignment has, in effect, inferred what

13. Notice that this ambiguity does not arise in the example presented in Figure 11 because there are no gaps in the sequence of '`yes`' and '`no`' symbols.

questions should have been asked in places where relevant information is missing and what the answers to those questions should have been.

Figure 12 about here

Figure 12

The single best alignment produced by SP61 with the patterns from Figure 9 in Old and the pattern '`Start 43 yes 44 7 yes no`' in New.

4.3.7 Information containing additions or substitutions

What about adding 'wrong' symbols to New or substituting 'wrong' symbols for 'correct' symbols? With respect to the pattern in New shown in Figure 11, adding '`no`' or '`yes`' at any point or substituting '`no`' for '`yes`' (or *vice versa*), will either lead to a different conclusion which is correct in terms of the modified version of New or, if the sequence of '`yes`' and '`no`' symbols does not correspond to any possible sequence in the decision network, then the results are unpredictable because of the many alternative ways in which partially 'correct' alignments may be formed.

The kind of 'errors' that the system can cope with most easily are ones which are clearly wrong such as, for example, the addition or substitution in New of symbols which do not match any of the symbols in Old. Provided there not too many of them, SP61 can bridge these kinds of error in much the same way as we saw in the example of 'spelling checking and correction' in Section 3.2.

As noted above, bridging gaps in information or by-passing inaccurate information is not the kind of thing that conventional decision networks or decision trees can do. Although this may be nothing more than a 'party trick' at present, there may turn out to be situations where these kinds of capability would be useful in a decision network or tree.

4.4 Rules

Expert systems typically use 'if-then rules' ('condition-action rules', 'condition-consequence rules' or, more briefly, 'rules') as the mainstay of their stores of knowledge although many expert systems (e.g., pioneering systems like MYCIN [1]) use rules in conjunction with other kinds of knowledge structure.

The patterns which may be used to represent a decision network or decision tree (Section 4.3) may be regarded as simple if-then rules where the 'yes' or 'no' answer to the question in the parent node is the condition, and a pointer to the corresponding grandchild node is the action or consequence. However, the kinds of rules which are used in 'mainstream' expert systems typically have features like these:

- The 'if' part of the rule may have a single condition but, typically, it has two or more conditions.
- In addition to the normal 'forwards' mode (with the 'if' and 'then' parts in their normal roles), rules are often used in a 'backwards' mode where the roles of the 'if' and 'then' parts are reversed.

- When there are two or more conditions, they are usually linked by an implicit or explicit 'AND' relationship but, in addition, other Boolean operators ('NOT', 'OR') are sometimes used.
- Associated with each rule, there is often some kind of measure of confidence in the conclusion which may be drawn from the rule. This confidence level is sometimes called a probability although, in many cases, the principles of probability theory are not followed strictly.

The subsections that follow discuss how these kinds of rules may be modelled in the ICMAUS framework. With the exception of the last feature - which, we shall assume, is accommodated by the probabilistic nature of the ICMAUS framework, I shall try to show how the features, above, may be modelled.

4.4.1 Why not replace rules with patterns?

If we abolish the structural distinction between the condition and consequence parts of a rule, then any number of conditions (linked by AND) and any number of associated consequences (also linked by AND) may be represented directly as a pattern as defined in Appendix A1 of [2]. Abolishing the structural distinction between 'if' and 'then' parts of rules makes sense when we consider that the distinction is already flouted when the left-to-right, if-then, direction of rules is reversed, as it often is.

The distinction between 'if' and 'then' may be eliminated from the structure of expert system rules but still retained within the system in the way patterns are processed. In the ICMAUS scheme, the 'if' parts of a pattern may be regarded as the parts which form hits (positive matches) with symbols from one or more other patterns, and, with respect to those other patterns, the 'then' parts of the given pattern are the symbols which have not been matched.

Removing the structural distinction between 'if' and 'then' paves the way for a system in which partial or 'fuzzy' pattern recognition is fully integrated with the processing of rules - with associated advantages of integration noted in Section 1.3.1 of [2]. Replacing rules with patterns has the additional advantage that, in most cases, a much larger range of 'if-then' relationships may be expressed. For example, if a rule like 'A B C $\rightarrow$ D' is treated as a pattern 'A B C D', the number of possible if-then relationships increases from 2 (one forwards and one backwards) to the number of possible subsequences in a sequence of 4 symbols (not counting the whole sequence) which is $((2^4 - 1) - 1) = 14$. This number increases exponentially with the length of the pattern.

In summary, representing rules as patterns in the ICMAUS scheme gives us a bonus of integration and flexibility and it enables us immediately to model the first two of the features listed above and one of the Boolean operators:

- A pattern can represent a rule with any number of conditions and any number of consequences.
- The flexibility of pattern matching in the ICMAUS framework means that the 'if' and 'then' parts of a pattern may, on different occasions, be left-to-right, right-to-left, middle-to-ends, ends-to-middle and many other possibilities.
- Since the symbols in a pattern may be regarded as forming an AND relationship without the need for an explicit Boolean operator, patterns naturally accommodate

this relationship amongst the elements of a rule.

4.4.2 Boolean relationships: OR

In expert systems and other systems which use rules, any rule that contains one or more OR relationships can always be replaced by rules that do not contain OR relationships. For example, a rule like this:

A OR B → C

can be replaced by two rules like this:

A → C
B → C.

More elaborate examples can be broken down in a similar way so that explicit OR relationships are eliminated.

If this is accepted, then there is no need to model OR relationships explicitly in ICMAUS. Any rule containing one or more instances of the OR operator may be translated into rules without that operator before the rules are converted into ICMAUS patterns.

4.4.3 Boolean relationships: NOT

No attempt will be made here to consider exhaustively how negation may be modelled in ICMAUS. Just one simple example will be presented here to suggest the general approach.

Figure 13 shows four patterns which may be placed in Old. The first two represent integer values ('1' and '0') for the symbols 'TRUE' and 'FALSE'. The boundary markers ('S' and '#S') are mnemonic for 'symbol'. The second two patterns represent a negation 'function' which converts '0' to '1' and *vice versa*. The role of the symbols ';' and '#S' in these patterns will be seen shortly.

```
S TRUE 1 #S                1
S FALSE 0 #S               1
not S 0 #S 1 #not          1
not S 1 #S 0 #not          1
```

Figure 13
A set of patterns used to illustrate how negation may be modelled in ICMAUS. For want of any better information about frequencies, each pattern has been assigned a frequency of 1.

Figure 14 shows two alignments formed by SP61 with the patterns from Figure 13 in Old and two different patterns in New. The first (a) is the best alignment formed with the pattern 'TRUE' in New. The second alignment (b) is the best alignment formed with the pattern 'not TRUE #not' in New.

Figure 14 about here

Figure 14

Alignments formed by SP61 with the patterns from Figure 13 in Old and two different patterns in New. (a) The best alignment formed with the pattern ‘`TRUE`’ in New. (b) The best alignment formed with the pattern ‘`not TRUE #not`’ in New.

In the first case, the best alignment delivers, in effect, a value of ‘`1`’ for the symbol ‘`TRUE`’. In the second alignment, this ‘output’ of the first alignment becomes, in effect, ‘input’ to the negation ‘function’ so that the value of ‘`TRUE`’ is converted from ‘`1`’ to ‘`0`’, thus, in effect, achieving the desired negation of ‘`TRUE`’.

4.4.4 A simple example of forwards and backwards ‘chaining’ in ICMAUS

Figure 15 shows, at the top, three ‘nonsense’ rules of the form ‘`IF`*x* ... `THEN`*y*’, where ‘*x*’ differentiates instances of conditions (‘`IF`*x*’) from each other and ‘*y*’ differentiates instances of consequences (‘`THEN`*y*’) from each other. The digit symbols which appear in these three patterns are provided to match corresponding symbols in the ‘framework’ pattern shown at the bottom of the figure. As in the example of ‘medical diagnosis’ described in Section 3.3, a ‘framework’ pattern is needed so that an alignment can be formed which can be unified without mis-matches of symbols of the kind described in Section 4.8.4 of [2], as we shall see in the example below.

```
1 IF1 2 7 IF2 8 16 THEN1 17                                 1
3 IF3 4 11 IF4 12 THEN1                                     1
IF3 IF4 13 IF5 14 19 THEN2 20                               1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20          1
```

Figure 15

A set of patterns representing a nonsense set of ‘if-then rules’, together with a pattern which provides a framework. As in Figure 13, each pattern has been assigned a frequency of 1.

When SP61 is run with the patterns from Figure 15 in Old and the pattern ‘`IF1 IF2`’ in New, the best alignment formed is simply an alignment between the pattern in New and the first pattern in Figure 15. This simple alignment may be interpreted as something like “Condition ‘`IF1`’ followed by condition ‘`IF2`’ means consequence ‘`THEN1`’” (we shall ignore the other symbols in the pattern from Old).

However, if we allow the system to go a little further beyond the evidence supplied, the second best alignment formed is the one shown in Figure 16. This alignment may be interpreted as a ‘forwards’ inference, followed by a ‘backwards’ inference, followed by another ‘forwards’ inference. These inferences may be described something like this: “From left to right, condition ‘`IF1`’ followed by condition ‘`IF2`’ means consequence ‘`THEN1`’. But, in the right-to-left direction, consequence ‘`THEN1`’ means condition ‘`IF4`’ preceded by condition 'IF3'. Finally, from left to right, condition 'IF3' followed by condition 'IF4' (but without condition 'IF5') means consequence 'THEN2'".

Figure 16 about here

Figure 16

The second best alignment formed by SP61 with the patterns from Figure 15 in Old and the pattern 'IF1 IF2' in New.

4.4.5 Chaining of inferences resembles the recognition of patterns with internal structure

It is not hard to see how, with some rearrangement of symbols, the medical diagnosis example (Section 3.3) might be interpreted in terms of the same kind of 'forwards and backwards chaining' of inferences as was described in the last subsection. For example, the symbol 'fever' might be placed at or near the right end of each pattern in which it appears so that the pattern could yield a 'left to right' inference like the first pattern in Figure 15. Then, with matches from 'fever' and some other symbols from New, the pattern which defines 'flu_symptoms' would yield a 'right to left' inference like the second pattern in Figure 15. And finally, if the symbols 'disease #disease' and the names of diseases were placed at or near the right ends of patterns, the pattern which defines influenza or the pattern which defines smallpox could, in the same manner as the third pattern in Figure 15, yield a 'left to right' inference the relevant disease.

5 Conclusion

This, the second of three articles about PrbRs as ICMAUS, has shown how the probabilities of ICMAUS inferences may be calculated and has then presented examples of PrbRs as it is manifested in best-match pattern recognition and information retrieval. The article ends with examples showing how the ICMAUS scheme can model PrbRs using networks, trees and rules.

The next and last article in this trilogy presents examples showing how ICMAUS may model PrbRs in hypothetical or 'what if' reasoning, in 'indirection' in information retrieval, in reasoning with default values, in nonmonotonic reasoning, and in solving geometric analogy problems. Finally, the article shows with an example how ICMAUS may provide an alternative to Bayesian networks for modelling the phenomenon of 'explaining away'.

References

1 Shortliffe, E. H. (1976). *MYCIN: Computer-Based Medical Consultations*, Elsevier, New York. Based on a PhD thesis, Stanford University, Stanford, CA, 1974.

2 Wolff, J. G. (submitted, a) Probabilistic reasoning as information compression by

multiple alignment, unification and search (I): introduction.

3 Wolff, J. G. (submitted, b) Probabilistic reasoning as ICMAUS (II): calculation of probabilities, best_match pattern recognition and information retrieval, and reasoning with networks, trees and rules (This article).

4 Wolff, J. G. (submitted, c) Probabilistic reasoning as ICMAUS (III): hypothetical reasoning, geometric analogies, default values, nonmonotonic reasoning, and modelling 'explaining away'.

5 Wolff, J. G. (submitted, d) Parsing as information compression by multiple alignment, unification and search. Two articles on which this paper is based ("Parsing ... SP52" and "Parsing ... Examples") may be obtained from http://www.sees.bangor.ac.uk/~gerry/sp_summary.html.

6 Wolff, J. G. (1988) Learning syntax and meanings through optimization and distributional analysis. In Y. Levy, I. M. Schlesinger and M. D. S. Braine (eds.), *Categories and Processes in Language Acquisition*. Lawrence Erlbaum, Hillsdale, NJ. Reprinted in Chapter 2 of Wolff (1991).

7 Wolff, J. G. (1982) Language acquisition, data compression and generalization. *Language & Communication*, 2, 57-89. Reprinted in Chapter 3 of Wolff (1991).