Cryptic crossword clue interpreter

M Hart and R H Davis

The paper considers the amenability of the cryptic crossword to automatic solving by computer. It is proposed that there are three stages involved in the solution of a cryptic clue: cataloguing of semantics, cataloguing of syntax, and actual solution. The solution stage can be further divided into four stages: keyword identification, syntax identification, clue manipulation, and solution checking. The study extends previous work in this area, which has already achieved the syntactic and semantic stages of solution, as well as the first (keyword identification) stage of solution. In an attempt to achieve the second stage of solution (syntactic identification), a program (PICCUP) is developed to parse and interpret cryptic clues. This it does with partial success, through use of a simplified cryptic crossword grammar.

programming languages, semantics, crossword compilation

Why computerize crosswords?

In recent years, many games and pastimes have been subject to some form of implementation on the computer. Where games such as chess, backgammon, and Scrabble are concerned, the reason for an effective implementation is simply to provide a human player with a skilled opponent[1]. As such it can be said that this aim has been generally successful. Indeed, in many cases the computer may prove to be a significantly superior player, due in the main to its superior information-processing capabilities.

There is an ongoing quest in the disciplines of artificial intelligence and knowledge-based systems to produce systems that are capable of imitating or performing in a similar (or better) way to humans, no matter what the task domain. A system that was capable of solving crosswords by using some form of reasoning other than 'brute force' search and pattern-matching techniques would certainly be true to the spirit of such investigations. Cryptic crosswords in particular would require an alternative method of solution, due to their lack of amenability to simple synonym-matching methods.

There is considerable scope for investigation into the automatic solution of cryptic crossword clues, the ultimate aim being to achieve actual solution. Before this point can be reached there are a number of key phases that must be passed through. These can be listed as follows:

(a) the semantic phase, i.e., cataloguing of keywords

(b) the syntactic phase, i.e., formulation of a crossword grammar

(c) the solution phase, which can be subdivided into:
  (i) identification of the keywords in the clue
  (ii) identification of the rule set being used by clue parser
  (iii) manipulation of clue interpretations as directed by the rule set
  (iv) checking solution against a thesaurus of synonyms

The first phase (a) has been fairly extensively documented — most regular crossword solvers would be capable of the compilation of such a catalogue. This phase is important in that it is the keywords in a clue sentence that dictate how it is to be solved. Without a knowledge of the keywords that a sentence contains, the subsequent grammatical analysis and solution would be all but impossible.

The second phase (b) — the formulation of a crossword grammar — is necessary because crossword clues are in effect a subset of natural English. As with any natural language, to solve problems in a crossword 'language', some common grammar must be understood. Understanding this grammar fully would allow any clue written in a common grammar to be solved. A partial understanding would allow only certain species of clue to be solved. For a computer to be capable of crossword solution (whether fully or in part), this crossword grammar must be formalized. One such formalization was carried out by Williams and Woodhead[2], as was the primary stage of the clue solution phase (the identification of possible keywords in the clue sentence). It is this clue solution phase (c) that will be of most interest in considering the present investigation.

Cryptic crosswords

The first modern crossword (or 'word-cross', as it was known originally) was composed by Arthur Wynne and appeared in the *New York World* on 21 December 1913. This differed from today's puzzles in that it was diamond-shaped and contained no blocked-in squares. During the early 1920s, crosswords became a craze in America, and it was during this time that crosswords began to assume their familiar form.

The first appearance of a crossword in a British publication was in *Pearson's Magazine* in February 1922 and the first *Times* crossword appeared on 1 February 1930[3]. British puzzles quickly developed their own style, being

Department of Computer Science, Heriot-Watt University, 79 Grassmarket, Edinburgh EH1 3HJ, UK

 0950–5849/92/010016–12

considerably more difficult than the typical American variety. In particular the cryptic crossword, as pioneered by 'Torquemada' (E.P. Mathers) for the *Observer*, became established and rapidly gained in popularity. Torquemada's successors, 'Ximenes' (D.S. Macnutt) and 'Afrit' (A.F. Ritchie), laid down what are generally considered to be the governing rules of the cryptic crossword. Smith and Steen[4] report that today crosswords appear in approximately 90% of the world's newspapers.

The general structure of the crossword is fairly universal. It usually consists of a (typically square) grid composed of empty and filled-in squares, known as the 'diagram'. The empty squares are arranged in intersecting rows and columns, known variously as 'lights' or 'slots', which are separated from each other by the filled-in squares. It is in these lights that the solver must place the words found by the solution of the clues. The starting squares of lights are indicated by a number, which also refers to the particular clue for a given light. As two clues may share the same number (one light lying horizontally and the other vertically), clues are referred to by both their number and direction ('across' or 'down'). In addition, the clue texts are usually separated into 'across' and 'down' categories to simplify their reading.

One crossword variant, which is peculiar to the UK, is that of the 'cryptic crossword', which is generally considered to be far more challenging than conventional crosswords. Although it shares the same grid structure and protocols as the conventional crossword, its clues are radically different. Each will usually consist of a sentence, which on first appearances may seem fairly abstract or surreal or may suggest an underlying theme. What is apparent, however, is that unlike a conventional crossword, it will not immediately suggest a solution (whether right or wrong), and a straightforward synonym-based approach to solution will be apt to fail. In fact, unless one knows how to analyse cryptic clues, the correct solution of such a crossword would be almost impossible to achieve.

The secret behind cryptic crossword solving lies not in reading the sentence literally, but in looking for keywords within the clue that indicate some operation that must be performed to arrive at a solution. There will usually be a synonym in the clue; the problem is knowing where to look for it. Only one section of the sentence will contain the synonym, while the remainder will contain instructions on how to manipulate words, parts of words, or letters to arrive at the actual solution. Thus a cryptic crossword clue can be solved in one of two ways. The first involves locating the probable synonym within the sentence and then trying to find a possible solution from it (much as is done when solving a conventional crossword). Once a contending word has been found, the instructions held within the rest of the sentence are carried out to determine whether this is, in fact, the correct solution. The alternative method of solution is just the reverse; the instructions are followed to arrive at a potential solution, which is then checked against the synonym. Purists would consider the latter method to be the technically correct method, as this would be the way the compiler would intend it to be solved, but in reality the former method is perhaps more likely to be employed.

An example of such a clue might be:

'Soil a broken heart(5)'

to which the solution would be 'earth'. This can be arrived at by the following route. The synonym (usually the first or last word or words in the sentence) may be one of 'soil', 'heart', or possibly 'broken heart' (it may also be the case that 'soil' and 'broken heart' are both synonyms of the solution; double-synonym clues are discussed later). However, a regular crossword solver would recognise the word 'broken' as a commonly used keyword indicating an anagram, in common with many words implying destruction or rearrangement. This type of keyword, indicating that an operation is to be performed, will be termed a 'pointer'. As pointers are usually adjacent to the words they refer to, this may mean that the solution is an anagram of either 'heart' (five letters) or 'soil a' (also five letters). If the anagram was of 'soil a', then the synonym would have to be of 'heart'. Alternatively, if the anagram was of 'heart' then it would have to produce a word meaning 'soil'. As it is, an anagram of 'heart' is earth, which is a synonym for soil. This, then, is the probable solution.

This example illustrates the second method of solution as described above. The reverse of this route (the former method) would be the identification of possible synonyms for soil, heart, etc., and then attempt to decide which one was correct by reference to the rest of the sentence. This example also serves to highlight the specificity of cryptic crossword clues; there is only one possible answer for this clue. Although there may be alternative synonyms that may fit the solution light (e.g., dirty is a five-letter synonym for soil), they will not be able to be derived from the 'cryptic' part of the sentence. In many (if not all) cases, a cryptic clue can be solved without any reference to the actual light into which it must fit (in terms of the words with which it must connect).

Although many cryptic clues have a similar elementary structure (that is, a synonym part followed or preceded by a cryptic, operational section), the anagram example outlined above is one of the simpler of the many forms of cryptic clue. Now some of the most commonly used varieties of clue are considered.

Simple cryptic clue

The most straightforward type of cryptic clue consists only of a definition of the solution. However, clues of this type can be some of the most frustrating, as the description invariably involves a play on words or a clever definition of the solution word or phrase. These clues do not contain any common keywords that would provide the solver with an idea as to the type of clue that they are faced with, or of how to proceed with its solution (indeed the compiler may include keywords from other clue types as part of the clue narrative to mislead the solver). Thus

it is left to the solver to attempt to understand the compiler's way of thinking and solve a clue to which there is no real 'algorithmic' method of resolution. An example of the simple cryptic clue is:

'Had a less important part in the orchestra?(6,6,6)'

The solution to this clue is 'played second fiddle'. As can be seen, the answer is arrived at almost by lateral thinking. It interprets a well known phrase in terms of its (almost) literal meaning. The question mark ? and the exclamation point ! are often used in such cases to denote this literalness.

Double-synonym clue

Along with the simple cryptic clue, the double-synonym clue differs from the majority of cryptic clue types in that it contains no obvious pointers to indicate what operation should be performed. Instead, the sentence will usually consist of two phrases or sets of words, which, while not synonyms of each other, are synonyms of the solution. These phrases may be separated by what can be termed an 'equality indicator'; a word implying that the two phrases are equivalent. Equality indicators include words such as gives, is, makes, and becomes. These words are also often used in other clue types to separate the synonym(s) from the rest of the clue. However, in most cases equality indicators will not be used and the two synonym phrases will run into each other, leaving the solver to decide where one ends and the next begins. An example of the double-synonym clue is:

'Unwarranted interference makes an argument.(5-4)'

In this case the answer is 'cross-talk', of which the first synonym is 'unwarranted interference' and the second is a pun on an argument. 'Makes' separates the two parts of the sentence and indicates their equivalence.

'Stone jar?(4)'

The solution to this clue would be 'rock', as both 'stone' and 'jar' are its synonyms.

Anagram clue

The anagram clue is perhaps the most widely used of clue types and is possibly one of the simplest to interpret. The structure of a simple anagram clue is generally as follows: the first or last word(s) of the clue are those of the synonym of the solution. Also in the sentence there will be a keyword or words indicating that an anagram is to be carried out (an 'anagram pointer'). Anagram pointers are typically words that imply destruction, rearrangement, or strangeness, and include examples such as broken, ruined, rebuilt, deranged, disorder, about, or madly. The rest of the sentence will contain the words or letters on which the anagram is to be performed. It may also be the case that there is an equality indicator (as in double-synonym clues) that separates the synonym part from the rest of the clue. Here is an example:

'Neural disorder is quite imaginary.(6)'

The solution here is 'unreal', as 'disorder' (the anagram pointer) indicates that an anagram is to be performed on 'neural' to produce a six-letter word that means 'quite imaginary'. Notice that 'is' is employed both to maintain the flow and grammar of the sentence and to equate the two halves of the clue.

'Employ Sue Turner.(3)'

The solution here is 'use'. This is because 'Turner' instructs that an anagram be performed on 'Sue' (i.e., to turn it) to produce an anagram of 'employ'. This example also serves to show the way in which a skilled compiler will produce a clue sentence that has a theme and meaning of its own. A laboured or ungrammatical sentence will alert the solver to its 'real' contents (i.e., the instructions that guide the solution of the clue). The task of the compiler is to distract the solver and lead them away from the solution by producing a sentence with more than one inherent meaning.

Reversal clue

The reversal clue is similar in many ways to the anagram clue, in that it consists of a synonym part, a pointer, and a set of words or letters (or their synonyms or shorthands) that are to be reversed to form the solution. It also may or may not contain an equality indicator equating the synonym with the rest of the clue. Reversal pointers include words suggesting reversal, overturning, or going backwards. For example, back, returning, over, retirement, round (also an anagram pointer) and, for 'down' clues, up. An example of the reversal clue is:

'Wolf goes up a stream.(4)'

In this case 'goes up' indicates that the word 'wolf' should be reversed (this is a 'down' clue) to produce a solution meaning 'a stream'. The solution is therefore 'flow'.

Enclosure clue

The enclosure clue involves the merging of two or more words so that one set (the word(s) indicated by the enclosure printer) encloses the other to form the solution word(s), which will be a synonym of the remainder of the clue. There is also another difficulty inherent in this type of clue. Consider a sentence where word x is to be split to enclose word y. It may be that the actual word x in the clue is used to enclose y, or it may be a synonym of x. The same is true of word y; it may be included in its given form, or else a synonym of y may be used. The pointers used to denote enclosure are, unsurprisingly, related to the physical act of enclosure or being split, such as sur-

rounding, enveloping, containing, holding, round (also an anagram and a reversal pointer), without, or about. Consider:

'Ford put certain points about publicity.(4)'

The solution here is 'wade', as certain points are 'w' and 'e' (see the section on shorthand notation, below) and are put 'about ad' (a synonym for publicity).

Insertion clue

The insertion clue is similar to the enclosure clue, except that in this instance the pointer indicates a word (or its synonym) that is to be placed inside another word (or its synonym) to produce the solution. Insertion pointers include words that suggest inclusion or envelopment, such as in, inside, surrounded, cutting, and held by. An instance of the insertion clue is:

'Writing material for Greek character in fireplace.(8)'

The solution is 'graphite', because a Greek character is phi, a fireplace is grate, and inserting the first into the second gives graphite, a writing material.

Hidden-word clue

The hidden-word clue, while not particularly common, is still worthy of attention, chiefly because of the way in which it uses the surface meaning of the sentence to distract the solver's attention away from the fact that the solution is lying before his or her eyes. This type of clue involves the concealment of the solution word (or less commonly, words) within a string of other words. The fact that this has been done is usually communicated indirectly to the solver by the hidden-word pointer, which can take one of two forms: it can be a word similar in meaning to the inclusion pointers described above (such as in, within, inside, etc.), which further complicates solution, or it can be a phrase such as we see or appears in. The solution is, of course, a synonym of the synonym part of the clue. An example of this clue is:

'Smack which appears in East Anglian ports.(4)'

The solution to this example is 'tang', a word meaning 'smack' (in the sense of 'taste'), and which is concealed in 'easT ANGlian ports'.

Juxtaposition clue

The juxtaposition clue involves the concatenation of synonyms, words, parts of words, and letters to form the solution. Juxtaposition may be indicated by a pointer such as and, with, by, meets, or next to, or it may be implied by the fact that two words are adjacent to each other in the clue sentence. Again, this clue type will have a synonym part, which may be separated from the rest of the clue by an equality indicator. This is one of the more common varieties of clue, and is certainly the largest user of 'shorthand' manipulations, which will be discussed later in this section.

For instance:

'To strike hotel messenger constitutes violent behaviour.(7)'

In this example, no juxtaposition pointers are used to separate the words that are to be concatenated (synonyms of 'to strike' and 'hotel messenger'). There is, however, an equality indicator separating the synonym part ('violent behaviour') from the rest of the clue. Thus ram + page = 'rampage'.

Compound clue

Although all the above-mentioned clues are commonly found as entire clues in their own right, on occasion two or more may be combined to form a compound clue. Where this happens, each clue operation will produce only a part of the ultimate solution. For example:

'Fodder for the right variety of grey donkey.(3-5)'

This gives 'rye-grass' as an answer. The operations that take place in the solution of this clue are: another word for 'donkey' is ass. 'Variety' is an anagram pointer indicating an anagram of 'grey' and, using shorthand notation (explained later), 'R' is an abbreviation for right. Therefore the solution is: r + yegr + ass, a synonym for 'fodder'.

Other clue types

There are numerous other forms of clue that are too obscure or rarely employed to merit detailed discussion, but which should none the less be mentioned. One such is what might be called the 'acronym' clue, as it involves the use of the initial letters of a series of 'dummy' words. When these letters are concatenated they will form a word or part of a word. This clue type is often used as part of a compound clue. Acronym pointers include words such as initially and capitally.

Another clue variant that is mainly employed within compound clues is that of truncation, whereby letters are removed from the beginning or the end of a word or a synonym of a word in the clue to leave a shortened version of that word (for example, stare could be truncated to leave star, or alternatively, tare). Truncation pointers include words such as reduced, short, little, or beheaded. Occasionally, the word Cockney will be used to denote that the H at the start of a word should be dropped (A Cockney hat would produce at, for example).

Also used in compound clues is translation, whereby an English word is placed next to a pointer such as French or German. This indicates that the English word should be translated into that language before being inserted into the solution. For example, The German would produce der, die, or das, and French bread would

give pain. In this way the crossword compiler has a source of word components that might otherwise be difficult to derive, while adding a theme to the clue.

Other clue types include the homophonic clue, where a pointer such as we hear, by the sound of it, or said indicates that a phrase in the clue should be treated as it sounds and not as it is written.

Occasionally, missing words from literary quotations are used as the solutions, in which case the form taken by the clue is similar to that described for the conventional crossword. The final variety of clue component to be discussed here is that of the cross-reference. This occurs when a number (in digits) is substituted for a word in the clue. This number refers to the answer to another clue in the crossword and usually will require that the other clue be solved before the current clue can be attempted.

Shorthand and abbreviations

As was demonstrated in the above section concerning compound clues, abbreviations of common words are frequently used in crossword compilation. The fact that a word is to be abbreviated is rarely stated explicitly in a clue. Instead, regular crossword solvers acquire a 'mental thesaurus' of commonly abbreviated words. As well as the more obvious shorthand forms of certain words, such as m for metre or us for America, there are 'traditional' words that regularly appear when a given letter or group of letters is required. For example, beginner, student, novice, or learner are all words that are used when the letter l (for learner) is required. Abstainer is used to denote tt, and point, direction, quarter, or way can be used to represent the letters that stand for the points of the compass. In addition, way can stand for the abbreviations of street (st) and road (rd). Other frequently used shorthands include kiss (k), pound (l), ring or hole (o), and key or note (the letters a to g). The roman numerals (i, v, l, c, d, and m) are represented by their respective numbers or by the more general words number or many.

PREVIOUS RESEARCH

Crossword compilation programs, due to their task domain's relative simplicity, are the most frequently implemented variety of crossword computerizations. The task of crossword compilation has been divided into three main problem areas[4]:

- grid or 'diagram' generation
- solution generation
- clue generation

Several authors have tackled the problems inherent in the first two stages. The majority have employed a high-level language approach[5-8, 14].

Davis and Juvshol[9] report a system (MAXWORD) that is capable of grid formation, light-filling, and clue provision. The package also allows computerized solution of conventional crosswords provided by the user, but as far as can be ascertained, a program has not been reported that can produce solutions for a cryptic crossword.

Cryptic crossword clue set construction, however, was considered by Smith and duBoulay[10], who were able to show that certain classes of cryptic clue could be generated automatically.

Williams and Woodhead[2] produced a system that employed a specially constructed 'language' (although more of a grammar definition) known as LACROSS (LAnguage for CROSSword puzzles). This language is capable of describing most, if not all, examples of cryptic clue. The aim of the system itself was to aid in, rather than actually carry out, the solution of cryptic crosswords. This system, unlike others, considered only one clue at a time without reference to the grid in which their solutions were to fit. The system analysed a clue sentence for keywords (such as broken to signify an anagram or learner to represent the letter l), which it would then list to the user with a brief explanation as to the effect that the word had in the clue. However, the development of the LACROSS grammar did not appear to be fulfilled by the accompanying program, which did not use the grammar to its fullest extent (being more concerned with semantics rather than syntax). This area would seem, therefore, to be an avenue worthy of further investigation, particularly in the use of the LACROSS grammar (or a subset of it) to provide, if not the solution, a detailed breakdown of given clues.

Keyword identification (as carried out by Williams and Woodhead) will in turn allow the secondary (grammatical analysis) step of the clue solution phase to be attempted. The aim of this study will therefore be to implement the second step of clue solution — the identification of the particular rule set that is being employed in any given clue. This will necessarily also involve the additional implementation of the first (keyword identification) step of the solution.

It must be stressed that such a program will not be capable of producing a completed solution to a clue, but rather a list of possible interpretations of that clue. This should then open the way for the execution of the final stages of clue solution either manually by the user or, at a later day, automatically.

The method by which the PICCUP (from Program for Interpretation of Cryptic Clues Using Parsing) program will carry out grammatical interpretation may be summarized thus:

- Scanning of a user-supplied clue to identify potential keywords such as pointers, shorthand, or ignored words. This will be carried out by comparing words in the clue with lists of keywords (and their meanings) stored on file, which will result in a list of possible keywords interspersed with unrecognised words. This stage will be similar in effect to Williams and Woodhead's program.
- Parsing the sentence recursively[11] to produce a set of valid clue interpretations.
- Displaying the interpretation of successfully parsed

clues as a summary that can be readily understood by the user.

As well as identification of keywords that are pointers, it was deemed important that the system should also be able to identify any shorthand words present in the clue. The main reason for this choice is that shorthand transformations are frequently used in crosswords and are often not immediately apparent. It would be useful to be able to inform the user as to the occurrences of potential shorthand words in the clue, as well as to list the possible abbreviations in each case.

Another class of word that it will be important to identify as a keyword is that of the ignored word. This type of word has examples such as perhaps, maybe, and then and has no purpose in the sentence other than making sure that the sentence reads properly or is grammatical. It will be necessary to ignore such words when parsing the clue, otherwise inaccurate or incomplete interpretations may be produced.

As well as the cataloguing of all such words it will also be necessary to derive and construct an overall crossword grammar before system implementation can be approached.

CROSSWORD GRAMMAR

Basis for grammar

Although there is little doubt that an overall crossword grammar is necessary for the successful solution of cryptic clues, the definition of such a grammar has a number of accompanying problems. As crosswords are compiled by a large number of different compilers for different audiences and for publications with different 'crossword traditions', it is highly likely that different dialects of cryptic crossword language exist. There is evidence to support this in the fact that regular crossword solvers often find that they become attuned to the style of clues in a given newspaper, or even those set by a given compiler (newspapers tend to use compilers in rotation). Therefore, if many differing dialects exist, can a definitive grammar be formalized?

Whether or not it is possible to produce an all-encompassing crossword grammar, what is probable is that the majority of clues are capable of being identified as belonging to one grammatical type or another. The following grammar was compiled by reference to Williams and Woodhead's previous formalization and by studies of British newspaper crosswords (taken mainly from the *Daily Telegraph, The Guardian, The Scotsman*, and *Today* newspapers).

This grammar is not intended to be a complete representation of all varieties of cryptic clue. Indeed, it is doubtful that the majority of common clues could be summarized using it. However, the intention is to demonstrate the feasibility of clue parsing, and thus the implementation of a totally comprehensive working grammar is beyond the scope of this study.

Clue rule sets

This section describes the rule sets that were finally decided on for use by the PICCUP program. The rule sets that have been derived for the clue sentence are necessarily limited. For a realistic crossword-solving grammar, the clue sentence could be identified as being one of a large number of clue types. In the event, it was decided only to deal with a subset of the possible clue types. The clue types that were omitted included:

- the simple-cryptic clue (which was not considered amenable to grammatical analysis)
- juxtaposition and compound clues (each of which would have needed especially large grammatical descriptions due to their numerous permutations)
- minor clue types such as translations and acronyms (which are often components in compound clues)

In addition to the pointers and materials required for each of the clue types, there are three other recognised types of keywords. These are:

- the equality indicator, which can be used to signify equivalence between the synonym part and the cryptic part of the sentence
- the shorthand word, which can be replaced by an abbreviated form of itself
- the ignored word, which is used to identify irrelevant words so that they can be excluded from the parsing process

Metasyntactic conventions

The metasyntactic notation (i.e., of the syntax of the syntax) that will be used to represent the derived clue rule sets is similar in many ways to the pseudo-Backus Naur Form (BNF) as employed by Williams and Woodhead[2]. The list of symbols that are used here is as follows:

→	= is composed of
.	= is followed by
/	= or
(x)	= x is optional
(x)*	= zero or more occurrences of x

The occurrence of the symbol word refers to the corresponding word in the clue, or else to one of its synonyms (which should be assumed to replace it in the clue; for example, canine could be replaced by dog). Note also that a sequence of words (i.e., (word)*) may also have a synonym that can replace them in the clue (for example, domestic animal could also be replaced by dog). The shorthand symbol indicates that a shorthand word is present in the clue sentence. Where this occurs, the shorthand word can be replaced by its abbreviation.

Clue sentence

The PICCUP crossword grammar is thus:

Clue → Double Synonym / Anagram / Reversal /
Enclosure / Insertion / Hidden-Word

```
Double Synonym → Synonym (.Equality Indicator).Synonym

Synonym → Word (.Word)*

Equality Indicator → Word (.Word)*
```

Figure 1. Rule set for double-synonym sentence

```
Anagram → Synonym (.Equ Indicator).AnagramSentence/
          Anagram Sentence (.Equ Indicator).Synonym

Anagram Sentence → Anagram Pointer.AnagramMaterial/
                   Anagram Material.Anagram Pointer

Anagram Pointer → Word (.Word)*

Anagram Material → Word (.Word)*

Synonym → Word (.Word)*

Equ Indicator → Word (.Word)*
```

Figure 2. Anagram rule set

```
Reversal → Synonym (.Equ Indicator).Reversal Sentence/
           Reversal Sentence (.Equ Indicator).Synonym

Reversal Sentence → Reversal Pointer.Reversal Material/
                    Reversal Material.Reversal Pointer

Reversal Pointer → Word (.Word)*

Reversal Material → Material Component (.Material
Component)*

Material Component → Word/Shorthand

Synonym → Word (.Word)*

Equ Indicator → Word (.Word)*
```

Figure 3. Reversal clue set

Rule sets are made up of the clue components as shown in Figures 1 to 6.

The rule set for the double-synonym sentence (where two synonym parts are either adjacent or separated from each other by an equality indicator) is shown in Figure 1. The anagram rule set (see Figure 2) allows for eight possible anagram forms, each of which contains an anagram pointer to indicate that an anagram must be performed on some anagram material to produce a solution that is a synonym of the synonym part of the clue sentence. The reversal clue (where a pointer indicates that the order of some material is to be reversed to arrive at a synonym of the synonym part of the sentence) is shown in Figure 3 and similarly admits eight possibilities. Figure 4 deals with the enclosure clue, which occurs

```
Enclosure → Synonym (.Equ Indicator).Encl Sentence/
            Encl Sentence (.Equ Indicator).Synonym

Encl Sentence → Encl Material.Encl Pointer.Incl Material

Encl Pointer → Word (.Word)*

Encl Material → Material Component (.Material Component)*

Incl Material → Material Component (.Material Component)*

Material Component → Word/Shorthand

Synonym → Word (.Word)*

Equ Indicator → Word (.Word)*
```

Figure 4. Enclosure clue set

```
Insertion → Synonym (.Equ Indicator).Insn Sentence/
            Insn Sentence (.Equ Indicator).Synonym

Insn Sentence → Insn Material. Insn Pointer. Encl Material

Insn Pointer → Word (.Word)*

Insn Material → Material Component (.Material Component)*

Encl Material → Material Component (.Material Component)*

Material Component → Word/Shorthand

Synonym → Word (.Word)*

Equ Indicator → Word (.Word)*
```

Figure 5. Insertion clue set

when a pointer indicates that one set of material is to be placed around a second set of material to produce a synonym of the synonym part of the clue. Likewise the insertion clue (see Figure 5) is similar to the enclosure clue, except that the pointer refers to the material for inclusion rather than exclusion. There are thus four possible enclosure forms of clue and four possible inclusion forms. Finally, the hidden-word clue involves the indication by the hidden-word pointer that a synonym of the synonym part of the clue occurs somewhere in the actual sequence of letters that forms the hidden-word material. There are eight hidden-word forms of clue as shown in the rule set in Figure 6.

SYSTEM

System overview

The PICCUP program performs four operations on any clue that is given. These operations are: clue input, keyword identification, clue parsing, and the display of any interpretations made.

Hidden-word → Synonym (.Equ Indicator)H-word Sentence/
H-word Sentence (.Equ Indicator).Synonym

H-word Sentence → H-word Pointer.H-word Material/
H-word Material.H-word Pointer

H-word Pointer → Word (.Word)*

H-word Material → Word (.Word)*

Synonym → Word (.Word)*

Equ Indicator → Word (.Word)*

Figure 6. Hidden-word clue set

Clue input

The first stage involves the acceptance of input from the user and the initial processing of the clue. The program prompts the user first for instructions (which will be displayed if the user requires them) and then for the clue. Once the user has entered the clue, he or she is asked to type in the total number of letters that the solution is to contain (this is to enable possible anagram clues to be identified, see Figure 7). The program will then pre-process the clue sentence. This involves separating the clue into its individual words. While this is being carried out, any punctuation is removed and capital letters are converted to lower case.

Identification of keywords

The next stage is to check each word in the clue sentence against the words held in the pointers and shorthand dictionary files to determine whether any of the clue words can be considered as keywords (pointers, equality indicators, ignored words, or shorthand words). If a word has more than one possible meaning (about, for example) each meaning is recorded. In addition, if the word is identified as being a possible shorthand word, all the possible shorthand versions of that word are read in from file.

Parsing clues

The next and major stage of processing involves the parsing of the clue. This is achieved by regarding the clue sentence as a list of words that still have to be parsed.

When a word has been parsed it is temporarily removed from the list and parsing resumes in a recursive manner. If there are no words remaining in the list this implies that identification of clue components is complete and it remains to check these against the rule sets.

If, on the other hand, there are still words remaining in the list, then the next word will be scanned to see if it can be considered as a clue component, namely, synonym, anagram pointer, anagram material, equality indicator, insertion material, enclosure material, hidden-word pointer, hidden-word material, or ignored word. For instance, to identify a word as an anagram pointer will require finding an entry in the pointer's dictionary. Anagram material, however, requires comparing the length of the present word with the length of the required solution. If the word is shorter than the solution, then the length of the next word in the list is also added (because anagram material can comprise a number of words). If the total number of letters is greater than the solution length, then this (or these) word(s) cannot be anagram material; shorthand words or synonyms of clue words are rarely, if ever, included in anagram material. If the number of letters in the scanned word(s) and the number of letters in the solution match, then these words are passed as anagram material and parsing continues for the remaining words in the list.

In much the same way, remaining clue components are considered; equality indicators, insertion and exclusion components, reversal and hidden-word components, and ignored words. Potential insertion, enclosure, reversal, and hidden-word materials are treated slightly differently to anagram materials. This is because these materials are not necessarily constrained by the length of the solution; insertion, enclosure, and reversal materials may employ shorthand versions or synonyms of their component words, while hidden-word materials have the only constraint that they must contain more letters than the solution.

Once clue components have been identified, multiple consecutive pointers or materials of the same type are replaced with an appropriate single pointer or material to reduce the parse to its briefest meaningful form. It is at this point that any identified ignored words are omitted and a comparison made against the rule sets. If no match is found, then the parse procedure is re-entered, otherwise, if a matching rule set is found, it can be considered that a possible interpretation has been made and this needs to be displayed to the user before a return to the parser routine. When the system can effect no further parsings, the parsing routine finishes with a report to the user that no (more) interpretations can be made.

Displaying interpretation

Two formats are used for the explanation of the program's interpretations. The first is simply a list of the clue words, alongside each of which is shown its description as a clue component (anagram pointer, reversal material, etc.). The second and perhaps more informative format is an English sentence that explains how the solution may be derived from the clue (assuming of course that this is the correct interpretation). For some clue types (inclusions, enclosures, and reversals), a list of shorthand versions of words identified as being both material components and shorthand words will also be shown.

A summary of the way in which PICCUP analyses cryptic clues is given in algorithmic form in the next section.

Figures 7 and 8 illustrate the results produced by a representative interaction with PICCUP. The solution to the clue 'Dare begin reforms – that's material' is 'gaberdine'. The program correctly notes that this is derived by performing an anagram on 'dare begin' (*Daily Telegraph* Crossword No 19, 709; 2 June 89).

PICCUP prompts the user to begin clue analysis by pressing the space bar. If no interpretations for the clue

```
**********************************************
                P I C C U P
**********************************************
Program for Interpretation of Cryptic Crosswords Using Parsing
**********************************************

Do you require instructions (y/n)? n

Type in the clue: Dare begin reforms – that's material.
How many letters in the solution (in total)? 9

Press <space bar> to begin analysis □
```

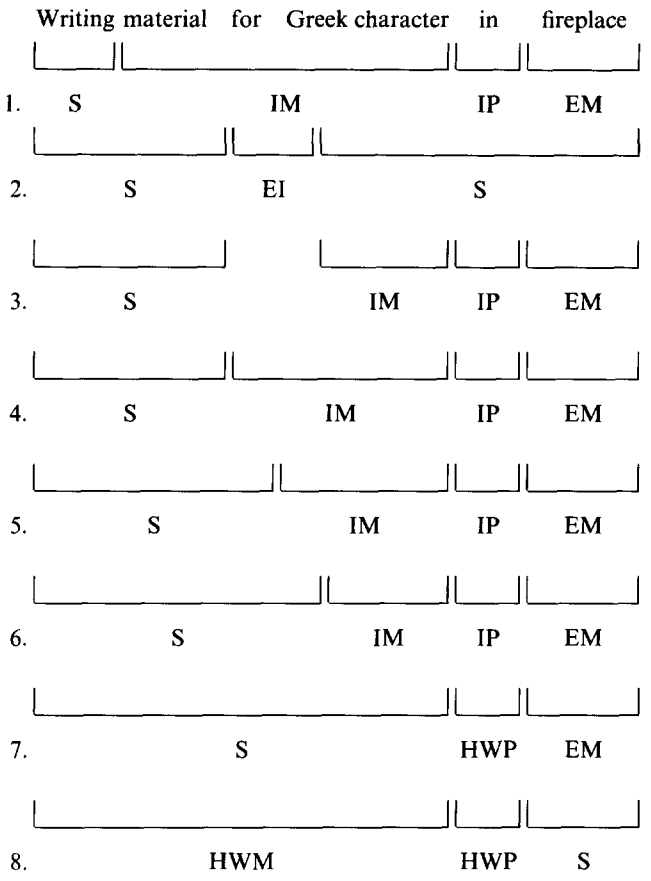
Figure 7. Results produced by representative interaction with PICCUP

```
**********************************************
                P I C C U P
**********************************************
Program for Interpretation of Cryptic Crosswords Using Parsing
**********************************************
INTERPRETATION * 1:

A possible interpretation is:
dare        – anagram material
begin       – anagram material
reforms     – anagram pointer
thats       – synonym component
material          – synonym component

Meaning:

REFORMS indicates that an anagram of DARE BEGIN
will produce a synonym of THATS MATERIAL.

Press <space bar> to continue analysis or <q> to quit.
No further interpretations can be made.
Any more clues to be analysed (y/n)? y
```

Figure 8. Results produced by representative interaction with PICCUP

can be derived, then the user is informed of this fact. Otherwise, the system will display its first interpretation and prompt the user to have the search for further interpretations continued.

For example, the clue 'Writing material for Greek character in fireplace' (*Guardian* Crossword No 18,506 and the section 'Cryptic crosswords') is parsed to match three rule sets (double-synonym, insertion, and hidden-word) and produce eight interpretations, thus:



where S, IM, IP, EM, EI, HWP, HWM are synonym, insertion material, insertion pointer, enclosure material, equality indicator, hidden-word pointer, and hidden-word material, respectively (see Figures 1, 5, and 6).

PICCUP algorithm

This section contains an algorithmic description of the procedures involved in the acceptance, parsing, and interpreting of a clue, as described in the previous section. It is written using an easily interpretable pseudo-code based on the C programming language. Braces ({}) denote that the grouping of procedures contained therein is to be considered as arguments of the preceding function.

```
begin
do {
    get clue from user
    count words
    identify all possible keywords
    copy clue to analysis list
    parse list {
        if no words remain in analysis list
        then if analysis matches grammar
                then display analysis and interpretation
                else return
        else { repeat if next word(s) of list is a clue component
                then remove word(s) from list
                        parse list
                        return word(s) to list
             }
while user has clues to be analysed
end
```

Implementation requirements

The choice of implementation medium was of considerable importance when designing the system. Prolog was a candidate for the software language that was to be used, due to its excellent symbolic manipulation and its goal-driven search strategies. However, it was felt that Prolog might prove to be too slow to deal with the large numbers of recursions that would be required of the system (a problem inherent in many high-level languages). Instead it was decided to use the C programming language for its speed, flexibility, and string-handling capabilities[13]. C has the additional advantage of being widely portable (being the mainstay of the Unix operating system). As C was to be used for the system's coding, the program was implemented on an Orion mainframe under a Unix environment.

PERFORMANCE

As may be seen from the example given in the last section and in Figures 7 and 8, the PICCUP program is able to provide interpretations for what can be less than obvious clues. However, for this example, it may be the punctuation and the 'surface' sentence structure that provides the degree of difficulty this clue possesses. In terms of crossword grammar, this is a fairly straightforward clue (as witnessed by the fact that it has only one interpretation). Problems arise when multiple interpretations are produced.

The program is capable of listing all (in its own terms) legal interpretations of a clue, but there will come a point for certain clues where the number of interpretations produced will be so great as to be useless to the user (the practical limit may lie at the point where the interpretations produced exceed six to seven in number). A large number of interpretations means that a large number of manual solutions must be attempted (the majority of which will be insoluble). The current method of reducing the user's workload involves the reading of the interpretations and the rejection of any that are noticeably incorrect (for example, the synonym components do not form a grammatical English word or phrase) and are obviously unsatisfactory. Now some of the responses given to a variety of clues are considered in an attempt to ascertain the practical limits of the program's performance. The examples that appear below are a selection of those already described or are as referenced. The accompanying descriptions are those given as interpretations by PICCUP. The asterisk has been added to indicate the correct interpretation.

A guide to use of PICCUP[13] is available on request, together with program listings.

Example 1

Clue: Employ Sue Turner.(3)
Solution: Use.
Interpretation 1:

> TURNER indicates that an anagram of SUE will produce a synonym of EMPLOY.*
> No further interpretations can be made.

Example 2

Clue: Hamlet picked for manner of derangement. (6,2,7) [*Daily Telegraph*, Crossword No 19,732]
Solution: Prince of Denmark.
Interpretation 1:

> DERANGEMENT indicates that an anagram of PICKED FOR MANNER will produce a synonym of HAMLET.*

Interpretation 2:

> FOR suggests that HAMLET PICKED and MANNER DERANGEMENT are both synonyms of the solution.

Interpretation 3:

> FOR suggests that HAMLET PICKED and MANNER OF DERANGEMENT are both synonyms of the solution.
> No further interpretations can be made.

Example 3

Clue: Getting on for ten scenes revised.(9) [*Daily Telegraph*, Crossword No 19,705]
Solution: Senescent.
Interpretation 1:

> FOR suggests that GETTING ON and TEN SCENES REVISED are both synonyms of the solution.

Interpretation 2:

> REVISED indicates that an anagram of TEN SCENES will produce (FOR) a synonym of GETTING ON.*

Interpretation 3:

> REVISED indicates that an anagram of TEN SCENES will produce a synonym of GETTING ON FOR.
> No further interpretations can be made.

Example 4

Clue: Wolf goes up a stream.(4)
Solution: Flow.
Interpretation 1:

> UP suggests that A STREAM (or its synonym), when reversed, will produce a synonym of WOLF.

Interpretation 2:

> UP suggests that A STREAM (or its synonym), when reversed, will produce a synonym of WOLF GOES.

Interpretation 3:

> UP suggests that WOLF (or its synonym), when reversed, will produce a synonym of A STREAM.*

Interpretation 4:

UP suggests that WOLF GOES (or its synonym), when reversed, will produce a synonym of A STREAM.
No further interpretations can be made.

Example 5

Clue: Engineers without love for caviar.(3)
Solution: Roe.
Interpretation 1:

FOR suggests that ENGINEERS WITHOUT LOVE and CAVIAR are both synonyms of the solution.

Interpretation 2:

WITHOUT suggests that LOVE (or its synonym), when enclosed by ENGINEERS (or its synonym) will produce a synonym of FOR CAVIAR.
Shorthand forms of ENGINEERS include, re.
Shorthand forms of LOVE include, o.

Interpretation 3:

WITHOUT suggests that LOVE (or its synonym), when enclosed by ENGINEERS (or its synonym) will produce (FOR) a synonym of CAVIAR.*
Shorthand forms of ENGINEERS include, re.
Shorthand forms of LOVE include, o.

Interpretation 4:

WITHOUT suggests that LOVE FOR (or its synonym), when enclosed by ENGINEERS (or its synonym) will produce a synonym of CAVIAR.
Shorthand forms of ENGINEERS include, re.
Shorthand forms of LOVE include, o.
No further interpretations can be made.

Example 6

Clue: Writing material for Greek character in fireplace.(8)
Solution: Graphite.
Interpretation 1:

IN suggests that MATERIAL FOR GREEK CHARACTER (or its synonym), when inserted into FIREPLACE (or its synonym) will produce a synonym of WRITING.

Interpretation 2:

FOR suggests that WRITING MATERIAL and GREEK CHARACTER IN FIREPLACE are both synonyms of the solution.

Interpretation 3:

IN suggests that GREEK CHARACTER (or its synonym), when inserted into FIREPLACE (or its synonym) will produce (FOR) a synonym of WRITING MATERIAL.*

Interpretation 4:

IN suggests that GREEK CHARACTER (or its synonym), when inserted into FIREPLACE (or its synonym) will produce a synonym of WRITING MATERIAL.

Interpretation 5:

IN suggests that GREEK CHARACTER (or its synonym), when inserted into FIREPLACE (or its synonym) will produce a synonym of WRITING MATERIAL FOR.

Interpretation 6:

IN suggests that CHARACTER (or its synonym), when inserted into FIREPLACE (or its synonym) will produce a synonym of WRITING MATERIAL FOR GREEK.

Interpretation 7:

IN suggests that a synonym of WRITING MATERIAL FOR GREEK CHARACTER is concealed somewhere in FIREPLACE.

Interpretation 8:

IN suggests that a synonym of FIREPLACE is concealed somewhere in WRITING MATERIAL FOR GREEK CHARACTER.
No further interpretations can be made.

CONCLUSION

Cryptic clues rarely contain many more than ten to 12 words, so that in parsing an average-length clue (six to seven words) the delay between the start of parsing and the display of the first (and subsequent) interpretations is scarcely noticeable.

The PICCUP system is capable of the interpretation of fairly sophisticated clue sentences, but in such cases is inclined to produce too many different interpretations. However, it can be considered to have progressed the unattempted second step of clue solution.

Although capable of interpreting many examples of the six clue types, there are a number of clues of each type that it will be unable to analyse correctly, due in part to the use of multiple-word pointers or to multiple representations of an individual word.

Should it be thought that crossword solution is still far beyond the reach of computing techniques, it should be noted that once a potential grammar has been identified, progress towards solution is to a large extent mechanical. The solving of anagrams, the search for synonyms, and the concatenation of words and letters to form new words are all machine-achievable tasks that can be

carried out with the aid of string manipulation tools, thesauri, and backtracking.

REFERENCES

1 **Turcan, P J** 'A competitive Scrabble program' in **Bramer, M A (ed)** *Computer game-playing, theory and practice* Ellis-Horwood (1983)

2 **Williams, P W and Woodhead, D** 'Computer assisted analysis of cryptic crosswords' *Computer J.* Vol 22 No 1 (1979) pp 67–70

3 **Augarde, T** *The Oxford guide to word games* Oxford University Press (1984)

4 **Smith, P D and Steen, S Y** 'A prototype crossword compiler' *Computer J.* Vol 24 No 2 (1981) pp 107–111

5 **Berghel, H** 'Crossword compilation with horn clauses' *Computer J.* Vol 30 No 2 (1987) pp 183–188

6 **Wilson, J M** 'Crossword compilation using integer programming' *Computer J.* Vol 32 No 3 (1989) pp 273–275

7 **Smith, P D** 'XENO: computer-assisted compilation of crossword puzzles' *Computer J.* Vol 26 No 4 (1983) pp 296–302

8 **Berghel, H and Yi, C** 'Crossword compiler-compilation' *Computer J.* Vol 32 No 5 (1989) pp 276–280

9 **Davis, R H and Juvshol, E J** 'Microcomputer compilation and solution of crosswords' *Microproc. Microsyst.* Vol 9 No 10 (1985) pp 500–506

10 **Smith, G and duBoulay, J** 'The generation of cryptic crossword clues' *Computer J.* Vol 29 No 3 (1986) pp 282–284

11 **Backhouse, R** *Syntax of programming languages: theory and practice* Prentice Hall (1979)

12 **Kernigham, B W and Ritchie, D M** *The C programming language* Prentice Hall (1978)

13 **Hart, M** 'PICCUP: a program for the interpretation of cryptic crossword clues' *MSc thesis* Heriot-Watt University, Edinburgh, UK

14 **Solomon, E** *Games programming* Cambridge University Press (1984)