Computer assisted analysis of cryptic crosswords

P. W. Williams and D. Woodhead

Computation Department, P.O. Box 88, Sackville Street, Manchester M60 1QD

An analysis of the structure of cryptic crossword puzzle clues is developed and a formal definition using a notation similar to BNF is provided. The language LACROSS seems to describe nearly all cryptic clues in British daily newspapers. The clues are made up of components of the solution and operators describing how the components are put together. A computer program is described which gives the possible interpretations of the words of the clue by using two main dictionaries, one of frequently used abbreviations and one of indicator words in the clues showing the possible operators and components. The complete BNF type definition of the LACROSS language is given and examples of output from the computer analysis program are presented.

(Received June 1977)

1. Introduction

Computer languages have one characteristic that clearly distinguishes them from natural languages. Any construction in a computer language is intended to have a unique meaning that can be automatically analysed by a computer program. By contrast natural languages are highly context dependent and a particular construction may well depend for its interpretation not only on the surrounding text but even on the cultural background of the speakers. It is quite common for the interpretation to depend on intangible variables such as the tone of voice or the rhythm of the delivery.

Crossword clues are a restricted form of English in which both the syntactic and semantic ambiguities in the language are deliberately exploited to give a variety of possible parsings for a clue. This limited form of English can be defined rigorously in a relatively simple form by using the Backus Naur form BNF in which some computer languages are defined. The language defined for the clues (called LACROSS, *LA*nguage for *CROSS*-word puzzles) is composed of elementary components and operators, from which more complicated forms are built up. For example in the phrase 'bad news' the text 'news' would be a component and 'bad' indicates the anagram operator. The words in the clue which indicate the operators will be called *signs*.

The representation of the allowable grammatical elements in the crossword clue language is complicated because the operators do not have a unique representation. The anagram operator is indicated by a few hundred different words or phrases, which necessitates a modification of the BNF notation. In this BNF formulation reference is made to the lists in which the possible words indicating operators are stored. Furthermore there are some signs which occur in more than one list and could represent different operators. Examples of this are 'of', 'in' and 'out'. Another complication is that the grammatical elements do not always come in the same order so that an operator may come before or after the component that it qualifies.

In crossword clues we have a grammar that is simple enough to be defined but which has considerable flexibility. It is this flexibility and the consequent ambiguity in parsing which makes crosswords a puzzle to solve rather than simply a logical exercise. This paper describes the language LACROSS which has been used to define the structure of crossword clues, and gives details of a computer program that has been written by Woodhead (1977) to assist a crossword puzzle solver. No attempt was made to provide a unique parse of the crossword clue for two reasons. Firstly, only a limited amount of time was available for the study and secondly if all the crossword clues are capable of being uniquely parsed then the compiler of the puzzle has failed to provide the expected mental exercise. However, further work might indicate methods of selecting the most likely parse for a clue in many cases, and such work could well have interesting applications in the field of automated language analysis.

The components of the grammar and the way it is represented are presented in Section 2 in a developmental manner and the formal definition is given in Appendix 1. The computer aids which have been developed are mainly based on the lists which have been established showing the relationship between the signs in the clue and the operators that they might represent. There are also lists of some of the components such as frequently used abbreviations, or the types present in a crossword clue from which an example must be selected. These aids are presented in Section 3, together with a description of the program which has been written to assist in parsing the clues.

2. Development of the grammar

2.1 *Notation for the clue analysis*

A clear method of representing the *structure* of the crossword clues is essential for developing a grammar. Various pieces of text are used as *components* which are combined by means of *operators* indicated by words in the text. These indicator words will be called *signs*. Some of these signs indicate *unary operators*; the signs may occur before or after the component. The situation is complicated by the fact that some operators are implied and no sign is present in the clue. It is convenient to represent the clue structure in terms of *compound components* generated from a unary operator combined with an elementary text component. This is particularly true if the operator is implied.

There are also *binary operators* which have a standard form with two components separated by a binary operator between them. However, as a result of the flexibility of the English language, the clue text may have the binary sign and the components arranged in several different orders. It is the uncertainty generated by this flexibility which makes clue solving an investigative activity rather than a mechanistic exercise.

The notation used here separates the operators and the components (elementary or compound) by using an asterisk in the clue structure for each operator and listing the operators in sequence after the clue structure. The text of the clue is given in quotation marks and equated with the *clue solution*. It is also useful to record the signs present in the clue text together with the operators that they indicate.

Brackets are used where necessary to indicate which part of a clue structure is affected by an operator, and an underline is used to show when components of a clue structure must be reversed to place them in the standard form for an operator. The symbol $\triangle$ is used for the fundamental definition of clue solution and occurs in nearly every clue structure. There is some ambiguity in the notation for many clue structures, which are essentially two different definitions of the clue solution. This ambiguity, however, does not create a serious problem in defining the clue structure.

2.2 *Examples of clue analysis*

The examples presented in this section will take the form 'Clue text' = clue solution, clue structure, sign-operator list. The last section may, of course, be empty.

2.2.1 *Simple definitions*

As a simple example we have a straightforward clue

'Personal charm' = amulet $\triangle$ (1)

A slightly more complicated example is a clue with two definitions of the same term

'Crab claws little ones' = nippers $\triangle = D$ (2)

This could be equally well represented as $D = \triangle$. The symbol D is used for a group of words which give a definition of the part of the clue solution. In this case the = operator is implied but in some clues there is a sign for the = operator, e.g.

'Highest speed for the A1' = top-rate
$\triangle = D$ (for, =) (3)

2.2.2 *Elementary components*

The source of the text to be manipulated comes, in elementary form, from three sources. There is a direct use of a piece of text from the clue represented by t; or from a literary quotation, represented by q; or from a cross-reference to another clue, represented by k. Small letters are used for these three, which are direct use of text without transformation. Examples of these are

'Hail to thee—spirit' = blythe q (4)
'Instrument of 5 down' = recorder $\triangle = k$ (of, =) (5)
(the 5 down clue solution was 'the law of the land')

2.2.3 *Unary operators*

Two very common unary operators are a for anagram and r for reverse as in the following examples.

'Get in odd bit of colour' = tinge
$t* = \triangle, a$ (odd, a) (6)

The asterisk shows that an operator has been used which is subsequently identified as an anagram operator and the sign used was 'odd'.

'Skilled Italian territorial detachment making a comeback' = able. $\triangle = D*, r$ (comeback, r) (7)

The anagram operator is occasionally implied as in

'How to raise up classical port' = Piraeus $*t = \triangle, a$ (8)

2.2.4 *Compound components*

The remaining unary operators are often more conveniently represented by combining the operator with the elementary component to give a compound component. This is particularly true of the operators s, x, l, d—which are often not represented by a sign. These are the abbreviation or shorthand operator s, the example operator x which selects one of a specified type, the like operator l which selects a synonym, and the operator d which takes a word defined by part of the clue. These generate compound components S, X, L, D.

Three other unary operators are m which takes the middle portion of a piece of text, i which takes the initial portion and c which takes the caudal or final portion. There are also v which is used to find a solution based on the sound of a clue component (i.e. verbal clue) and z which is used for the translation of a clue component. These give compound components M, I, C, V, and Z. Some examples of these components in clue structures are given below. The binary operator j for juxtaposition is necessary for some of these.

'Discharges a bill and then goes' = acquits
$\triangle = S*D, j$ (then, j) (9)

'Lower a foundation' = abase $\triangle = t*L, j$ (10)

'River bird presiding over pop session' = dee-jay
$X*X = \triangle, j$ (x, river; x, bird) (11)

'Stop in this abode, sister' = desist $\triangle = M$ (in, m) (12)

'Bury at the end of winter' = inter $\triangle = C$ (end, c) (13)

'River smell, say' = oder $\triangle = V$ (say, v) (14)

'Number of a French novel' = roman
$\triangle = Z$ (of, = ; French, z) (15)

The reason for using these compound elements is clear from a complete structural analysis of the following clue.

'Rejects and acts with N. Ireland county' = turns down
$\triangle = (t*)*(t*), ljx$ or $\triangle = L*X, j$ (with, j) (16)

The second analysis using compound components is much simpler and gives greater clarity for clues with binary operators or more complicated constructions.

2.2.5 *Binary operators*

The juxtaposition operator j is often implicit. It is convenient to introduce it to give a more readable clue structure as in this example.

'Make a mathematical proposition of the gold printers measure' = theorem
$\triangle = t*S*D, jj$ is used rather than $\triangle = tSD$ (17)

The problem with the binary operators is that there are several different orders in which the standard form may appear. There are two components G_1 and G_2 to be combined by a binary operator $*$. In the standard form G_1*G_2 the operator primarily qualifies the first component G_1. However, the flexibility of English allows the same order to imply that the operator primarily qualifies the second component. We denote this by $\underline{G_1*G_2}$ where the underline signifies that the order must be reversed before the clue is semantically interpreted, i.e. $\underline{G_1*G_2} \rightarrow G_2*G_1$.

This is exemplified by this example of the operator e for enclosing

'March round on a sovereign' = monarch
$t*t = \triangle, e$ (round, e) (18)

contrasted with an example for the operator f for the final part of a clue solution.

'Exultant cry of BBC doctoı finishing work' = whoop
$\triangle = \underline{X*S}, f$ (of, = ; finishing, f; work = op, S) (19)

The binary operator primarily qualifies the first component in example 18 so that 'march' enclosing 'on' is in the correct order since march is the component which encloses. However, 'work' as a final part to 'BBC doctor' is the correct standard order in example 19 since 'op' comes at the end of the clue. There are four other changes in order which may be discovered on parsing and which need to be altered to standard form before semantic interpretation. These are described in the grammar in Appendix 1 and examples are given in 20 and 21.

'The girl he embraces is Greek' = Hellene
$\underline{X(t*)} = \triangle, e$ (embraces, e) (20)

the standard order is 'he' enclosing 'the girl'.

'In witchcraft fabric one can see through is attractive' =
magnetic $\underline{(*L)\ D} = \triangle, o$ (in, o) (21)

The standard order is 'net' outflanked by, (o), 'magic'.

Another binary operator is w standing for the first component without the second component.

There are two binary operators, b when G_1 comes before G_2 and f when G_1 follows G_2 that are sometimes used as unary operators. In that case we have G_1*, b or $*G_1, b$ showing that G_1 comes at the beginning of the clue solution, and $G_1*; f$ or $*G_1, f$ showing that G_1 comes at the end.

Examples of the binary operators are

'Achieving an examination success in advent' =

compassing $\triangle = D*L, o$ (in, o) (22)

'Mature set infiltrated by two cardinals' = grown-up

$\triangle = L*S, e$ (infiltrated by, e; cardinals = WN, S) (23)

'Wear certain lines after some hesitation' = erode

$\triangle = L*S, f$ (after, f; hesitation = er, S) (24)

'Man has a follower in Canada' = Alberta

$\underline{X(t*)} = \triangle, f$ (x, man; follower, f) (25)

'Row started over a short skirt in Italy' = Rimini

$I * (S*I) = \triangle, b\,j$ (started, i; over, b; $a = i$, S) (26)

'Unsuitable in a pot lacking nothing' = inapt

$\triangle = t*s, w$ (lacking, w; nothing = o, S) (27)

This grammar can be extended, as described in the appendix, to cope with the more complicated clue structures but there are occasional clues which do not easily conform to the defined grammar. The following example requires some contortion to transform it to standard form.

'Can leg look broken' = glance

$\underline{t(= \triangle)*}, a$ (broken, a) (28)

However, complicated examples can be easily represented, e.g.

'Bad king about to eat at a point thats very sharp' =

Knife-edge $((*t)*L)*S = \triangle, aej$

(bad, a; about, e; point = e, S) (29)

'The reply the artist sent back contained bad news' = answer

$\triangle = (S*)*(*t), rea$

(sent back, r; contained, e; bad, a) (30)

3. The computer program

The computer program takes a clue and suggests possible interpretations of the words it contains. No attempt is made to select from these alternatives so if a word has more than one alternative all of these are printed out. The program is arranged so that new clues can be used to update the system at any time as new signs are noted. The program was written by D. Woodhead based on a simpler definition of the grammar in Appendix 1.

3.1 *The Components of the Program*

The program consists of three parts.

1. A group of dictionaries containing various lists which will assist in recognising significant parts of the clue.
2. The clue analysis section which accepts a crossword clue, extracts the separate words and prints out the possible significance of these words as recorded in the dictionaries. The program also contains counters which increment each time a word is observed, so that a statistical analysis of a complete crossword puzzle or a series of puzzles can be obtained.
3. An editing section which allows items in the dictionaries to be amended, deleted, added, or examined.

3.2 *The sign dictionary*

Each sign has a number of words which clue compilers use to indicate the type of component or how they are to be assembled in the solution. The entry with the most signs is undoubtedly the anagram operator, a, with hundreds of words in its list but the operators e and o which place one piece of text inside another also have long lists. All the lists for the different operators are merged into a single list which is stored in alphabetical order. Where a word has different interpretations all these are stored with the word. In this sign list each entry includes the word and the interpretation of the word in symbolic notation.

3.3 *The abbreviation dictionary*

There are many traditional words in crosswords that have well known abbreviations, e.g. 'Abstainer' = tt, 'direction' = n, e, s, w etc. These are stored in alphabetical order together with the abbreviations.

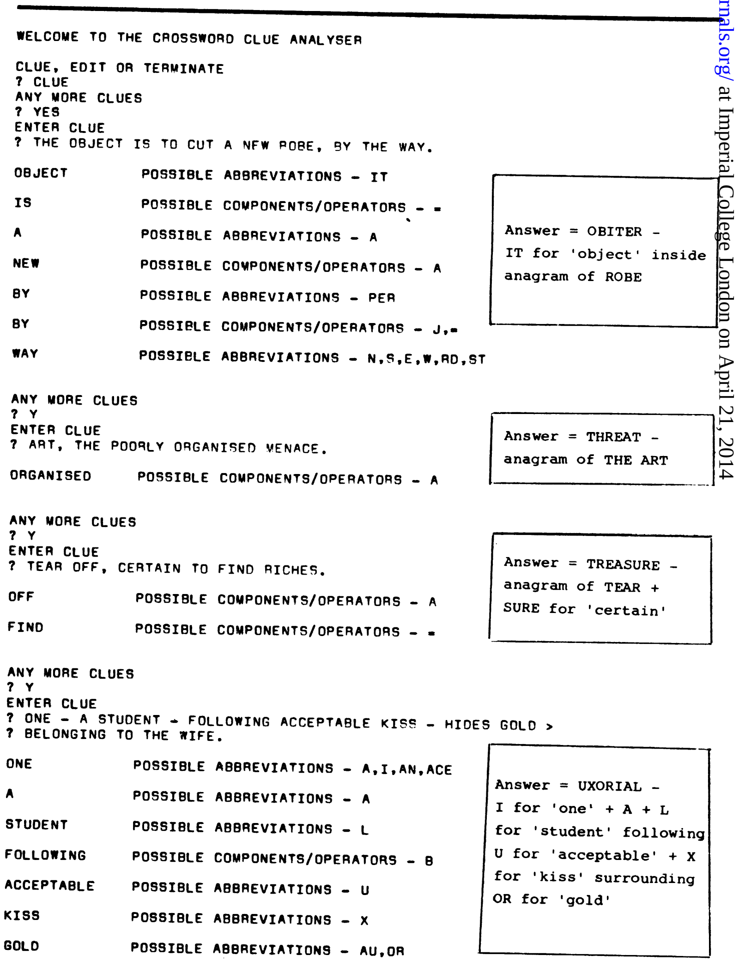
3.4 *The examples dictionary*

This is rather different from tne other dictionaries since it lists the text on which the x operator is likely to be operating. The number of possibilities if an X component is present in a crossword clue is very large and it would be quite unrealistic to store these. The purpose of the examples dictionary is to suggest when to consider an example as a component of the solution.

3.5 *The clue analysis section*

This section isolates individual words and puts them in the same format as the dictionary entries. A comparison is then made with the dictionary entries and when a match is found the word is printed, together with an indication of the type of item it might be, and the interpretation of the word. The printout in **Fig. 1** shows some typical output from the system.

The clues have definitions and solutions as follows:



```
WELCOME TO THE CROSSWORD CLUE ANALYSER

CLUE, EDIT OR TERMINATE
? CLUE
ANY MORE CLUES
? YES
ENTER CLUE
? THE OBJECT IS TO CUT A NFW ROBE, BY THE WAY.

OBJECT         POSSIBLE ABBREVIATIONS - IT

IS             POSSIBLE COMPONENTS/OPERATORS - -

A              POSSIBLE ABBREVIATIONS - A

NEW            POSSIBLE COMPONENTS/OPERATORS - A

BY             POSSIBLE ABBREVIATIONS - PER

BY             POSSIBLE COMPONENTS/OPERATORS - J,-

WAY            POSSIBLE ABBREVIATIONS - N,S,E,W,RD,ST
```

Answer = OBITER -
IT for 'object' inside
anagram of ROBE

```
ANY MORE CLUES
? Y
ENTER CLUE
? ART, THE POORLY ORGANISED MENACE.

ORGANISED      POSSIBLE COMPONENTS/OPERATORS - A
```

Answer = THREAT -
anagram of THE ART

```
ANY MORE CLUES
? Y
ENTER CLUE
? TEAR OFF, CERTAIN TO FIND RICHES.

OFF            POSSIBLE COMPONENTS/OPERATORS - A

FIND           POSSIBLE COMPONENTS/OPERATORS - -
```

Answer = TREASURE -
anagram of TEAR +
SURE for 'certain'

```
ANY MORE CLUES
? Y
ENTER CLUE
? ONE - A STUDENT - FOLLOWING ACCEPTABLE KISS - HIDES GOLD >
? BELONGING TO THE WIFE.

ONE            POSSIBLE ABBREVIATIONS - A,I,AN,ACE

A              POSSIBLE ABBREVIATIONS - A

STUDENT        POSSIBLE ABBREVIATIONS - L

FOLLOWING      POSSIBLE COMPONENTS/OPERATORS - B

ACCEPTABLE     POSSIBLE ABBREVIATIONS - U

KISS           POSSIBLE ABBREVIATIONS - X

GOLD           POSSIBLE ABBREVIATIONS - AU,OR
```

Answer = UXORIAL -
I for 'one' + A + L
for 'student' following
U for 'acceptable' + X
for 'kiss' surrounding
OR for 'gold'

Fig. 1 Sample output from the computer program

$S*(*t) = \triangle, oa$	obiter	(31)
$t* = \triangle, a$	threat	(32)
$(t*) * L = \triangle, aj$	treasure	(33)
$((S*t*S) * (S*S)) * S = \triangle, jjfje$	uxorial	(34)

3.6 *The editing section*

If new clues generate further entries for the dictionaries, or if errors are discovered, then the editing package gives the opportunity to amend the system at any stage. For example, the first clue contains the sign 'cut' indicating the *o* operator which is not yet in the dictionary, the final clue contains the sign 'hides' which indicates the *e* operator.

4. Summary and conclusions

The formal definition of the LACROSS language for representing cryptic crossword clues appears to be suitable for nearly all clues in the British daily newspapers. Considerable assistance in solving clues can be given by the computer program which informs the solver of the possible interpretations of the words in the clue. It is often the case that solving a clue is inhibited by an early misinterpretation of the clue, which the solver insists on pursuing. Since the system is adaptive it is intended that the dictionaries will be updated with new entries.

There are interesting possibilities for developing the work. By combining the identified components and operators it would be possible to give a list of possible parses. Because of the flexible nature of both the semantic interpretation of the signs and the binding of operators to operands, this is a substantial exercise. It does, however, seem to be a halfway house between formal computer languages and the considerable freedom of natural language, which could profitably be studied.

Analysis of the clue definitions which occur in published crosswords show that there are some combinations that are frequently used and others that rarely occur, although they appear to be quite feasible constructions. This exposition of the formal structure will enable crossword puzzle compilers to consider unusual combinations of operators to produce further variety.

Appendix 1 BNF definition of grammar

$\langle\text{clue}\rangle ::= \triangle/\triangle = \langle G\rangle/\langle G\rangle = \triangle$
$\langle G\rangle ::= \langle Y\rangle/\langle H\rangle/\langle N\rangle$
$\langle Y\rangle ::= t/k/q$
$\langle H\rangle ::= \langle U\rangle\langle G\rangle/\langle G\rangle\langle U\rangle$
$\langle N\rangle ::= \langle G\rangle\langle P\rangle\langle G\rangle/\underline{\langle G\rangle(\underline{\langle P\rangle\langle G\rangle})}/(\underline{\langle P\rangle\langle G\rangle)\langle G\rangle}/$
$\underline{\langle G\rangle(\langle G\rangle\langle P\rangle}))/(\underline{\langle P\rangle\langle G\rangle)}\ \langle G\rangle/\langle G\rangle(\underline{\langle G\rangle\langle P\rangle})$
$\langle U\rangle ::= s/x/l/d/m/i/c/z/v/a/r/f/b$
$\langle P\rangle ::= e/o/w/j/f/b$
$s ::= \phi/s_i:s_i\ \varepsilon\ \{\text{slist}\}$
$x ::= \phi/x_i:x_i\ \varepsilon\ \{\text{xlist}\}$
$l ::= \phi/l_i:l_i\ \varepsilon\ \{\text{llist}\}$
$d ::= \phi/d_i:d_i\ \varepsilon\ \{\text{dlist}\}$
$m ::= m_i:m_i\ \varepsilon\ \{\text{mlist}\}$
$i ::= i_i:i_i\ \varepsilon\ \{\text{ilist}\}$
$c ::= c_i:c_i\ \varepsilon\ \{\text{clist}\}$
$z ::= z_i:z_i\ \varepsilon\ \{\text{zlist}\}$
$v ::= v_i:v_i\ \varepsilon\ \{\text{vlist}\}$
$a ::= a_i:a_i\ \varepsilon\ \{\text{alist}\}$
$r ::= r_i:r_i\ \varepsilon\ \{\text{rlist}\}$
$e ::= e_i:e_i\ \varepsilon\ \{\text{elist}\}$
$o ::= o_i:o_i\ \varepsilon\ \{\text{olist}\}$
$w ::= w_i:w_i\ \varepsilon\ \{\text{wlist}\}$
$j ::= \phi/j_i:j_i\ \varepsilon\ \{\text{jlist}\}$
$f ::= f_i:f_i\ \varepsilon\ \{\text{flist}\}$
$b ::= b_i:b_i\ \varepsilon\ \{\text{blist}\}$

Notation

1. The underline is used wheie the natural order of the text must be reversed to put the elements into standard sequence. Brackets are used to indicate where elements must be taken as a unit.
2. For each of the operators in the sets *U* and *P* a list of words that indicate the type of operator are stored. These are called *signs*.
3. It is convenient to extend the BNF grammar to give more components by defining $S ::= s\langle G\rangle/\langle G\rangle s$, and similarly for all the operators *s*, *x*, *l*, *d*, *m*, *i*, *c*, *z*, *v*. We can then look upon the grammar as a combination of operands
$$t/k/q/S/X/L/D/M/I/C/Z/V$$
and operators
$$a/r/e/o/w/j/f/b\ .$$
4. In order to create a readable format, the operators are shown by asterisks in the clue analysis and then listed in sequence after the analysis.

Semantics

The various symbols in the grammar have the following meaning.

$\triangle$	is the reference to the meaning of the clue.
$\langle G\rangle$	is the cryptic part of the clue.
$\langle Y\rangle$	is the piece of text which is manipulated by the operators.
$\langle H\rangle$	is a component formed by using a unary operator.
$\langle N\rangle$	is a component formed by using a binary operator.
t	is a piece of the text in the clue.
k	is a piece of text from a cross-reference to another clue.
q	is a piece of text from a quotation.
$\langle U\rangle$	is a unary operator. The operators *s*, *x*, *l* are often implied and there is no sign for them in the clue text.
$\langle P\rangle$	is a binary operator which links two components. There are several formats in which the operators appear and they are discussed below.
$\langle S\rangle$	is a shortened form of a word or phrase.
$\langle X\rangle$	is an example of the type given in the clue.
$\langle L\rangle$	is a synonym for a clue component (*L*ike).
$\langle D\rangle$	is a word defined in the clue.
$\langle M\rangle$	is the middle section of a word or phrase.
$\langle I\rangle$	is the initial section of a word.
$\langle C\rangle$	is the final or caudal section of a word.
$\langle Z\rangle$	is a translation of a component.
$\langle V\rangle$	is a word which sounds like the clue description (*V*erbal).
$\langle A\rangle$	is an anagram of the clue component.
$\langle R\rangle$	is a component reversed in order.

The lower case letters corresponding to the above eleven components indicate the operators which create them.

The binary operators appear in standard format as $\langle G\rangle\langle P\rangle\langle G\rangle$ and this will be represented in the analysis as
$$\langle G\rangle * \langle G\rangle \qquad (\langle P\rangle)$$
They have the following meaning

$\langle G_1\rangle\ e\ \langle G_2\rangle$	means G_1 enclosing G_2.
$\langle G_1\rangle\ o\ \langle G_2\rangle$	means G_1 outflanked by G_2.
$\langle G_1\rangle\ w\ \langle G_2\rangle$	means G_1 without G_2, i.e. with G_2 taken away.
$\langle G_1\rangle\ j\ \langle G_2\rangle$	means G_1 juxtaposed to G_2. This is a very common operator which is often implied, with no sign to represent it. It is used in the notation to separate operators and operands in a neat readable fashion.
$\langle G_1\rangle f \langle G_2\rangle$	means G_1 placed at the end. G_2 may be omitted giving a unary operator.
$\langle G_1\rangle\ b\ \langle G_2\rangle$	means G_1 placed at the beginning. G_2 is sometimes omitted, giving a unary operator.

Reference

WOODHEAD, D. (1977). Computer Aids to the Solution of Cryptic Crossword Puzzles, Third Year Project Report, UMIST.