Does he analyse a tragic stammer? (14): towards a grammar of cryptic English

Abstract

This paper investigates the possibility of arriving at an exhaustive grammar for a sub-set of visible English, namely CE (cryptic English, the language of cryptic crossword puzzles). It proposes two levels of analysis: (1) a general syntactic structure for the sequencing of top-level clue-parts (synonym, operator, crypto-mechanism) and (2) a more detailed typology for the sub-syntax of crypto-mechanisms. It is proposed that, with some minor exceptions, this two-level description comes close to furnishing such an exhaustive grammar.

Does he analyse a tragic stammer? (14): towards a grammar of cryptic English

Alec McHoul, Murdoch University, Western Australia*

0. Introduction

Nowhere, perhaps, is the fact that writing is much more than just written-down-speech — that visible language can have an existence outside mere translations into talk — more clearly shown than in the case of crosswords. Crosswords are a ubiquitous part of everyday visible-language use and form a veritable sub-set of English itself — and, moreover, one that is routinely accomplished by the manipulation of visual clues (see 2.2 below for a minor exception). What remains to be accounted for, however, is the exact grammar of this important component of visible English.

Grammars of entire natural languages are notoriously difficult to achieve for a number of reasons: the fact that no one really knows whether a finite number of rules can generate an infinite number of sentences (let alone more complex utterance-units); the fact that there appears to be an element of chaos to natural languages (such that their use can change the rules for their use); the fact that natural languages are historically unstable and subject to unpredictable changes in the forms of life which they both anchor and serve; and so on. However, it does seem feasible to construct a grammar for limited sub-sets of a natural language, its jargons, codes, slangs and argots. One linguistically interesting case in point is cryptic English (CE), the sub-language used to construct and solve cryptic crossword clues. In this case, two facts are evident which do not necessarily apply to any natural language. (1) CE works with a remarkably fixed lexicon by virtue of agreements between users about standard dictionaries — in most cases the agreement falls upon <u>Chambers</u> as the preferred option (because of its policies of combining British and US English spellings and of writing all words out in full rather than resorting to abbreviations). In principle, this lexicon can be expanded by the inclusion of well-known proper nouns such as the names of countries, authors and historical figures. (2) Clue structure (the equivalent of sentence structure in a natural language) is strictly determined by a limited set of syntactic rules; "lower"-level components (such as graphology), as well as "higher" level components (such as semantics), when relevant to CE utterances (such as in cases of homography/homophony and synonymy) actually function syntactically in CE. In this sense, CE is highly determined by the LEXICON + SYNTAX combination — a state of affairs which is far from true of any natural language. The lexical component can be articulated fairly easily, then; but, so far the syntactic rules have not been clearly described — and it is the purpose of this paper to do just that. Once these are supplied, we will have achieved something closely approximating a comprehensive grammar of CE.

0.1 Grammar and pragmatics

There is a slight caveat to the otherwise clear principles announced above. The grammar of CE is not quite exhausted by the formula, LEXICON + SYNTAX — though this is only trivially the case. A complete statement of clue syntax (plus the standard lexicon) almost entirely accounts for the pragmatics of crossword writing and reading (or, more correctly, compiling and solving) — but not quite. In fact, CE is almost unique with respect to the tremendous degree of convergence between its sheerly formal grammar and the pragmatics of its use. Put another way, the ideal compiler-solver could be described almost entirely in terms of his/her competence in the writing and reading of CE's clue syntax were it not for the sheerly pragmatic fact that clue-solutions also comprise "lights". That is, they have to be fitted on to a grid of light-coloured squares, half-symmetrically arranged around intervening "darks". This means that there is an added pragmatic dimension to compiler-side competence: the capacity to fit items from the lexicon into a (usually 15 x 15 square) grid such that all intersecting lights have commonly intersecting characters. Moreover: it means that there is, in addition, an added pragmatic dimension to solver-side competence: the capacity to relate clue syntax to letter-intersectional probabilities (a capacity akin to those used Scrabble® and cloze tests). These additional pragmatic factors need not concern us here. They are much more crucial to another language community, that of the Quick-Crossword (or General Knowledge Crossword) — but they have little bearing on CE since, at least in principle, a light should be derivable from the formal clue grammar (LEXICON + SYNTAX) regardless of any intersecting letters available at the time the light is derived.

1. Basic syntactic structures

The primary level of CE syntactic structure is given by the following rule (**r1**): all CE clues can have, in principle, three components. These are: a light synonym [S], a crypto-mechanism [C], and operator [O]. We must write "in principle" rather than "in practice" because, in practice, the last-mentioned of these elements [O] can be null [Ø]. So to rephrase the primary rule (**r1.1**): all CE clues must have a light synonym [S] and a crypto-mechanism [C] and they may, in addition, have an operator [O].

Any light synonym will, rather obviously, be a synonym of the light. Any crypto-mechanism will be an encrypted means of achieving the light for which the light synonym is a synonym. Any operator will connect (and therefore distinguish between) [S] and [C]. Hence we can initially parse an example of a CE utterance (clue) as follows:

[Little boy pinches ring,]	[producing]	[utter confusion] (5)
[C]	[O]	[S]

Light = CHAOS (Times 4318)[1]

Here the operator tells us that [C] is literally "producing" a synonym of [S]. Examples of standard (perhaps clichéd) primary operators are: is, 's, gives, gets, shows, makes (and their variants). Less common examples are: contains, in, as, gives rise to (and their variants). In a large number of cases, however, the operator is null:

[Relish] [boat getting in winning position] (7)

[S] [Ø] [C]

Light = KETCHUP (Times 4305)

The null operator, in this case, could be replaced by any positive filler which generated roughly Standard English syntax (is, producing, and so on). But in such cases of null operators, solvers have the additional problem of separating [S] and [C] and compilers have, as it were, the advantage of greater ambiguity as to the exact parsing of the clue. In addition to positive and null operators, there are also cases of implied operators:

[To be or not to be] [—] [rewritten in quotes] (8)

[S] [O?] [C]

Light = QUESTION (Times 4327)

Here the implied operator might be that is (the).

Coming on to the contents of the synonym components: these can be relatively straightforward matters (though see 2.3 below) — so that Relish generates KETCHUP, utter confusion generates CHAOS, and so on. However, as the Relish case shows, parsing (by compilers or solvers) has the additional complexity that the part of speech suggested by the clue syntax may not be (and frequently is not) the same part of speech as required in the light. CHAOS and utter confusion are both nominal; but Relish (in Relish boat getting in winning position) is either adjectival or verbal while KETCHUP is nominal. In some instances, CE synonyms can reasonably stretch the limits of Standard English (SE) synonymy. Hence last places to train can generate

TERMINI (Times 4319) or, even more obscurely, His can generate GREETINGS (Times 4317).

((*General note on CE-SE relations.*: Such peculiar forms of synonymy in combination with part-of-speech shifts (between clue and light) point to what is effectively a proto-rule concerning CE syntactic constructions. That is, clues should take the form of SE phrases or sentences which, when read in SE (rather than as CE forms), have very little to do with the light. In fact, when read in SE, they can appear to stretch the limits of ordinary syntax and to utterly ignore semantic constraints. Colourless green ideas sleep furiously would (were it good CE, which it isn't) be a perfectly acceptable utterance. So CE clues read in SE are deliberately misleading. However, this proto-rule is by no means universal, as the case in 2.6.4 below shows.))

The contents and structures of crypto-mechanism components (C-parts) are yet more subtle than those of the primary-level synonyms (S-parts) — so much so that they will be the subject of a further section below. In fact, crypto-mechanisms contain a sub-grammar of their own and effectively generate the distinctiveness of (the somewhat limited set of) clue types — as we shall see.

What remains to be done for now is to specify the primary syntactic types of clues. This requires a second primary-level syntactic rule (**r2**): any clue's synonym [S] must fall in either theme or rheme position. Now while there are certain circumstances in which this rule can be circumvented (particularly in the "double duty" cases discussed below, 2.6.4), this is extremely unusual (or else indicates poor compiler competence). So, given **r1** and the almost-complete operability of **r2**, the possible primary level clue types allowed by the syntactic component of CE are as follows:

1) SOC

2) COS

3) OCS

4) SCO

and allowing for null operators, the following types will also be grammatical:

5) SC

6) CS

Hence the set is effectively limited to six types.[2] All six are common and plentiful in standard CE, with the possible exception of type 4) where the operator falls in rheme position, which is much rarer. However, the following is arguably a case:

[Tether] [that is restricting ram?] [Just the opposite] (3,2)

[S] [C] [O?]

Light = TIE UP (Times 4317)

In this case we could parse the clue as follows: the operator connects and divides [S] and [C] by telling us that a synonym for tether (verb) "is just the opposite" of that is (letters I and E, from i.e.) restricting (going around the outside of) ram (TUP). Hence the word for ram (TUP) now has to go around the outside of IE, generating TIE UP. However, another parsing would show the main operator to be null and take that is restricting ram? Just the opposite as a free-standing crypto-mechanism. Either parsing (SC or SCO) is possible, showing that CE can contain syntactic ambiguities at the primary level — thereby turning upon deliberately and cryptically misleading parsing — so that it is often unclear to solvers, prima facie, which of the types 1) to 6) above is actually in play in any given case. This normal and routine play is utterly expectable in CE.

Accordingly, compilers are expected to play with primary SOC operations in two ways: by purposefully ambiguating parsing and by (as we have seen) shifting parts of speech between clues and lights. Hence one possible solvers' maxim for primary-level operations would be the following: discern all possible parsings and decide the most likely of these on the grounds of expectable speech-part exchanges between the clue's light synonym and the possible light. Thus, given a clue of the following kind:

Noble artist with old car in 1 ac (10) (Times 4318)

solvers could initially read 1 ac as the synonym (where 1ac = CHAOS, see above). However Noble is a much more likely candidate since it's much more prone to speech-part shifts from the adjectival to the nominal than its competitor. As it turns out, this is the correct, though finally surprising parsing.

[Noble] [artist with old car in 1 ac] (10)

[S] [Ø] [C]

Light = ARISTOCRAT (Times 4318)

Note that, in this case, in 1ac (that is, in chaos) is in fact an anagram indicator, or anagram sub-operator as it will be called below (2.4.1).

2. The sub-syntax of crypto-mechanisms

So far we have dealt only with primary-level considerations of the basic SOC structure of CE syntax. However, there is a further sub-syntax particular to crypto-mechanisms themselves and these, in effect, generate the possible array of clue-types. That is, what makes clue-types distinctive is not so much their rather limited primary-level SOC arrangements as it is their C-part possibilities. I suspect that these possibilities are finite and can be listed, at least for very standard forms of CE such as one may find in Times puzzles. However, there are two caveats here too. (1) The clue-types do permit a degree of latitude or play and so we may find that, historically, the list of types is subject to addition and deletion. (2) The clue-types given below are pure or ideal types; for, in practice, nearly all of them can be mixed with others (with the double synonym being a notable exception) — yet, as we will see, in CE, ideal types do not have to be analytically derived from surface phrase structures — they can exist empirically, on the surface, as actual clues. With these two caveats in mind, however, I believe that the following is a fairly accurate and exhaustive list of possible (ideal) and, as it turns out, actual (empirically instanciable), C-part structures.

2.1 The light inclusive[3]

The light inclusive is perhaps the simplest of crypto-mechanisms. It involves no more and no less than the inclusion of the light itself in the C-part. Hence:

[Placed in barrel, it]['s] [fired] (5)

[C] [O] [S]

Light = RELIT (Times 4318)

Here the C-part actually contains the letters of RELIT: Placed in barREL IT. There is a strong convention here (though by no means a strict rule) that light-inclusive C-parts should signal themselves with sub-operators — that is, operators acting inside the C-part itself (by contrast with primary-structural operators which connect and distinguish S-parts from C-parts). In the case in point, this sub-operator is Placed in. Examples of light-inclusive C-part sub-operators are: within, contains, inside, holding, involving and so on. Notice the similarity between these and primary-level operators: they clearly can and do get used as sources of deliberate clue-type ambiguity or misleading parsing.

((*General note on the status of sub-operators*: Light inclusives, as we have seen, routinely need sub-operators to signal them; but they are not alone in this respect; for in fact, almost all C-part types, with some obvious exceptions (such as the double synonym) can use sub-operators. These can signal the subsumption of one string by another (inside, around, containing, etc.); the addition of one string to the beginning of another (above, over, on, etc.); addition to the end of another string (below, under, with ... finally, etc.); the reversal of a string (about, around, vice versa, etc.); an anagram of a string (see 2.4.1); a homonym of a string (see 2.2); as well as a number of other specialist operations (2.1 and 2.1.1 being cases in point). While there is clearly an intimate connection between these sub-operators and the C-part types they serve (and hence between them and overall clue types), I do not want to argue that sub-operators alone can define clue types. This for two reasons. (1) There is no necessary congruence between sub-operators and C-part types. For example, inside can signal subsumptions of any combination of anagrams, synonyms and homonyms (and it can also signal the light inclusive); therefore, it is the anagramic, the synonymic and the homonymic etc., which are the genuine markers of C-part types (rather than their operators). (2) Relatedly, there is considerable ambiguity among operators. For example, about can signal a reversal, an anagram or a subsumption (of, for example, synonyms), as well as itself being a stock synonym for re. and sc.))

2.1.1 Mixed inclusives

A variation of the light inclusive is the mixed inclusive. Here the C-part contains the light but not directly. Instead the light is given by the first letters of a sequence of words within the C-part (marked by sub-operators such as initially or primarily); by the final letters of a sequence (marked by finally or lastly); or by the odd-numbered or even-numbered letters in a C-part sequence (marked by sub-operators such as evenly and oddly). Hence:

> Primarily something that you press to injured chin (7) (Times 4322)

generates the light STYPTIC. Note in this case too, a clever use of double duty (2.6.4): the light-synonym is utterly coextensive with the component of the C-part which is working initially/inclusively. A nice variation on the mixed inclusive is the following:

> Will soon get repetitions eliminated for the President (6)
>
> Light = WILSON (Times 4317)

That is, Will soon will generate WILSON if its repeated letters, l and o are eliminated. This extends the mixed inclusive to a point where it is, perhaps, hybridised with the eliminator sub-type (2.5.1).

2.2 The "we-hear"

As a small concession to the phonic capacities of writing, one kind of C-part, the "we-hear" is based on homophony. It involves the following structure: HOMOPHONE + SUB-OPERATOR or, inversely, SUB-OPERATOR + HOMOPHONE. An example is:

```
[Raised] [[money] [for auditors]] (4)
   |        [HOM]      [SUB]
   |            \       /
  [S]              [C]
```

Light = BRED (Times 4318)

Decoded, the primary parsing along with the C-part sub-syntax reads: find a word that sounds like a synonym for money but is actually a synonym for Raised. The imperative ("find a word that sounds like") is given by the sub-operator (the actual "we hear" part, for auditors) and examples of these are: we hear (rarely used because of its obviousness), in audition, sound[s], reported, and so on. Hence:

Football striker reported (6)

Light = SOCCER (Times 4321)

2.3 The (sub-)synonym

In this section we are going to examine the structure of C-parts that involve sub-synonyms and which, when pieced together, generate a light which is (of course) synonymous with the light synonym. Before moving on to these types, a general note is in order on the nature of CE synonyms — be they light synonyms (S-parts) or sub-components of crypto-mechanisms (C-parts).

((*General note on synonymy:* While CE synonyms may be straightforward — such that almost no speaker of standard English would object to them — they may also be peculiar to the sub-language of CE itself and known only to insiders. For example, standard abbreviations (and the rest) are almost always used in conjunction with other C-part elements — usually, again, synonyms but also with anagrams and the rest. Very occasionally they occur in isolation:

Head teachers' organisation (3)

Light = NUT (Times 4317)

In some ways these "standards" are controversial since they don't easily pass through international boundaries, while crosswords themselves do. The better the crossword, the more likely it is to be taken up by overseas publications or, more recently, electronically via the internet; but the risk is that the standard abbreviations don't travel as well or as easily. Although cryptic crosswords originated in the USA, the well-travelled ones today originate in the UK, particularly in England — and a rather middle-class southern England to boot, where compilers are often men and seem to be very knowledgeable about military and sporting matters — not to mention the game of bridge, smatterings of Latin, Greek, Spanish, French, German and Italian, as well as English and Classical literatures. Hence, solvers are expected to know that <u>artist</u> generates the letters RA (Royal Academy); that <u>soldiers</u> generates the letters RE (Royal Engineers) or REME (Royal Electrical and Mechanical Engineers) with the occasional concession to GI (Government Issue); that <u>sailors</u> generates RN (Royal Navy) or ABs (Able Seamen) or else the argot-based synonyms <u>tar</u> and <u>jack</u>; that <u>queen</u> is most likely to be ER; that <u>college</u> will be King's or Queen's and that <u>school</u> will be Eton (if not plain old <u>sch</u>.); that <u>honour</u> means CH (Companion of Honour), if not a high card in bridge; that <u>decoration</u> is OBE, MBE or CBE; that <u>drink</u> can be the sea; that <u>miners</u> are the NUM; and perhaps even that <u>terrorists</u> are the

IRA. Many of these stock abbreviations have now passed into CE lore, such that many solvers recognise them without much referral to their originals. And, surely enough, they're to be found in the lexicon (Chambers). Less common are the -er argot synonyms which have grown up around CE itself and which are bewildering to CESL (CE as a Second Language) speakers. Regarded now as corny and clichéd by many, they still appear from time to time. Based on the idea that nouns ending in -er can look very much like nominalised verb-forms with -er suffixes, the main examples are:

flower => river (since it flows)

bloomer => flower (since it blooms)

butter => ram (since it butts)

lower => cow (since it lows)

master => ship (since it has masts)

adder => mathematician or calculator (since he, she or it adds)

number => anaesthetic (since it numbs)

worker or social worker => ant or bee (since they are social creatures — hymenoptera — with workers among their number)

With this caution in mind, we can now pass on to the various sub-types of synonym-based clues)).

2.3.1 The double synonym

Routinely, double synonym clues consist simply of two words — hence they are pure clue-types which cannot be hybridised with others. However, their primary-level parsing is technically ambiguous: if the words are synonymous, it matters not which counts as the light synonym and which as the crypto-mechanism. Either will do duty for the other; and being two-word clues, they are bound to respect **r2** (light synonym in theme or rheme position). However, double synonyms have the advantage (for solvers) of instant recognisability since they are (almost but not quite) the only possible clue-types with two (and two only) words.[4] Hence:

[Exchange] [vehicle] (5)

[S/C] [S/C]

Light = TRUCK (Times 4318)

Note that double synonym clue-types (in order to retain their cryptic element) frequently, in fact almost without exception, involve speech-part shifts. In the above example, <u>vehicle</u> can hardly shift in this way, but <u>Exchange</u> can (being possibly nominal, verbal or adjectival). Since <u>vehicle</u> only has a nominal form, then <u>Exchange</u> must be acting nominally towards the light.

2.3.2 The 2+

The double synonym is very straightforward, then, but it can be complicated by one of its sub-types: the two-plus word (2+) double synonym. In this variation, at least one of the synonyms consists of two or more words and, for this reason, it can occasionally contain a primary-level operator. Whether or not it does, it is at this point (and, even more so, at the next, 2.3.3) that solvers can expect the possibility of indecent puns, especially from inexperienced compilers. Highly competent CE compilers, however, can reach almost poetic realms with the 2+ double synonym (and, as a sign of self-conscious good manners, the exclamation mark or the question mark may be included):

One noted for longevity and plenty of bottle! (10)

Light = METHUSELAH (Times 4318)

Or else, a very skilled compiler can manage to keep an entirely prosaic feel to the clue, resisting the temptation to either the poetic phrase or the indecent pun, and so disguising crypticity with simplicity:

Exceedingly good example of a marine mammal (5)

Light = WHALE (Times 4318)

How chess players regard each other, in general (6,3,5)

Light = ACROSS THE BOARD (Times 4317)

2.3.3 The single synonym (pun)

The single synonym is exceedingly rare in standard dialects of CE since it is almost always simply a pun, and so, in actual cases, the question mark is obligatory. (In fact further signs of dubiety can be included here, such as perhaps?. Thus the clue His, perhaps? generates the light, GREETINGS (Times 4317).) In this type of clue there is effectively no crypto-mechanism (C-part) at all on the surface of the clue; instead the crypticity is given by a high degree of disguise in the synonym:

Intended to put the band here? (4,6)

Light = RING FINGER (Times 4323)

((*General note:* Any crossword which uses more than a single instance of this type risks the charge of being in poor taste — though this is a matter of communicative politeness rather than formal grammar.))

2.3.4 The mixed sub-synonym

This variation is very close to its parent type since it has the following structure: a genuine light synonym plus a C-part constructed out of two (or, occasionally more) sub-synonyms. Hence

[Party — reason] [for] [needing lots of food] (8)

[C] [O] [S]

Light = RAVENOUS (Times 4321)

Here, the C-part contains two sub-synonyms: Party generates RAVE, while reason generates NOUS; hence the case is semantically straightforward and syntactically linear. In other cases, the two (or more) synonyms have to be further operated on in order to generate the light:

[Forceful] [detectives holding crowd back] (7)

[S] [C]

Light = DYNAMIC (Times 4320)

In this case, the further operations on the sub-synonyms, detectives (CID) and crowd (MANY) are signalled by the sub-operators, holding (one sub-synonym contains the other in the light) and back (the sub-synonyms have to be read in reverse). A simpler form is:

Unsophisticated old city in African province (7)

Light = NATURAL (Times 4320)

This evidence suggests that there may be a further clue type, the subsumption (***2.3.5**), where one string is inserted inside another to achieve the light, and signalled by sub-operators such as in and holding (as in the above cases). The matter is far from clear, however, since almost all cases of subsumption involve the subsumption of synonyms (UR for old city; NATAL for African province; and so forth). In this case they should be classed as mixed sub-synonyms. However, there are cases where, for example, anagrams are involved (with synonyms) in the subsumptional arrangement:

Early, perhaps — coming round a day before now (7)

Light = ALREADY (Times 4325)

The parsing here is very tidy and linear: Early, perhaps => anagram of Early; coming round a day => subsumption into the foregoing of synonym for a day => ad, by initialisation; before now => light synonym. So there is evidence for **2.3.5** but, more economically, this type could be looked at as a synonym-anagram hybrid (section 3 below) managed in part by a subsumption sub-operator. Subsumption may then be a sub-operator type rather than a strict clue type. (See above, 2.1, for a general note on the distinction between clue types and sub-operators.)

2.4 Anagrams

2.4.1 Straight anagram

The C-part of straight anagrams is routinely structured as follows: ANAGRAM + SUB-OPERATOR (or vice versa). The sub-operators basically signal that "this is an anagram" and their numbers in CE are legion. They include: circled (from the fact that solvers may write anagrams in circles in order to remove their misleading linearity), wrong and wrongly, awry, badly, mixed, turned, alter(ed), around, (re)organised, changed and changing (and their variants). Anagram sub-operators can, however, get very obscure. Indeed they can be any term which is roughly synonymous with rearrangement, including thrashing, milling, beating, pulping and so on; as well as loosely, difficult, new, and special. ((Readers of this paper might like to decide whether the word analyse could be such a candidate.)) Whatever the case, the massive expansion of anagram markers may have to do with the fact that anagrams are rather easy to spot by the distortions of Standard English syntax they can lead to — hence they need to be further disguised by new and ingenious sub-operators. Where the anagramic C-part is extremely obvious, compilers may even drop the sub-operator altogether, saying, in effect, "if you need help to see this one, you must be blind". Again, the more skilled compiler will be able to keep the sub-operator and still disguise the anagram by neatly seguing it into the overall syntax of the clue. Here is a fabulous case in point:

> What better hopes for changing those in authority? (3,6,4,2) (Times 4312)

In this case, for once, I won't give the light and simply bemoan the fact that those in authority contains 16 rather than 15 letters!

2.4.2 The part anagram

This is a very common C-part type in CE. While the more specialised types such as the light inclusive and the "we-hear" are restricted (at least by good compilers) to one-per-puzzle, part anagrams abound. The conceit here is that part of the light is given anagrammatically while the rest is given by some other means, for example by synonymy. Hence:

> Distressing detail about broken down nag (8)
>
> Light = POIGNANT (Times 4312)

Here the light synonym is Distressing and the C-part works by asking us to take the letters from nag (anagram sub-operator = broken down) and putting a synonym for detail (point) around them (subsumption sub-operator = about).

2.5 Addition and subtraction

2.5.1 The eliminator

Eliminative C-parts work by asking us to find a particular word, dropping one (or occasionally more) of its letters and thus making another, the light. Sometimes the elimination will be the first letter of the word, so that the elimination sub-operator will be a term like headless or beheaded. Sometimes it will be the last letter of the word, so that the sub-operator will be docked, bottomless, and the like. A creative example would be:

> Most effective medicine stops onset of fatal attack (8)
>
> Light = LAUGHTER (Times 4317)

Here the sub-operator is stops onset and the term for elimination is a synonym of fatal attack (slaughter). The eliminator can also have a partial rather than total use in a clue — as with the anagram and the synonym, it can be mixed. Hence:

2.5.2 The mixed eliminator

Usually, the eliminator, as with the anagram, is mixed with a partial synonym in order to complete the C-part — though it can be mixed with other types. Here's a case in point:

In this situation, say, having eliminated a dissenting belief (6)

Light = HERESY (Times 4312)

Note that the sub-operator, here, can be very literal (eliminated) because what is to be eliminated is so well hidden: it's the single letter a! So a sub-synonym (In this situation / Here) plus an eliminator (say minus a) generates the light, HERESY (synonym: dissenting belief).

2.5.3 The expander

The expander is simply the opposite of the eliminator. It means adding one letter (or occasionally more) to a word in order to get the light. That letter can be added inside, at the start of, or at the end of the initially derived word in order to expand it. Again the letter to be added is usually the first of another word (marked by sub-operators such as initially). Alternatively it can come from the middle of the word (marked by sub-operators such as heart and centre). And lastly, it can come from the end of the word (marked by sub-operators such as tail and finally). In the case of the expander, the expansion component can be more than a single letter (which is rare in the case of the eliminator) — though it's rarely more than two. A rather nice instance is:

Dog with quivery tail is mine (8)

Light = COLLIERY (Times? memory)

As with the eliminator, the expander can form the whole C-part in its own right or it can be mixed with others. Hence:

2.5.4 The mixed expander

By this stage the character of the mixed expander should be obvious. A case in point should easily display the structure:

Study new score including Tchaikovsky's first musical pieces (9)

Light = CONCERTOS (Times 4317)

Also by this stage, then, it can be seen that C-parts can be parsed in a number of ways. I like to think of this as a mixed expander because its most prominent element (it seems to me) is the nice play with Tchaikovsky's first (as it turns out, the letter T). However, it's also a part anagram (score => CEROS) with a part synonym (Study => CON) prefixed. A further consequence of this kind of hybridisation (see section 3, below) is that the eliminator can exist in equal measure, in a clue, with the expander. In this case it just might have status as a clue type in its own right (***2.5.5** the replacer):

Slightly indecent and arrogant, changing leader (7)

Light = NAUGHTY (Times 4326)

Replacing the leader (first letter) of haughty (slightly indecent) with another, n, we get NAUGHTY. But this is rare and, as noted above, may not have genuine status as a clue type sui generis.

2.6 The rare and the special

2.6.1 The shift split

Many of the above examples show that lights are often derived by special operations on C-parts. One such special operation is the shift split. Its essence is that the crypto-mechanism (C-part) generates two-or-more words which, were they split in a different place, would have the same meaning as the light synonym (S-part). A very good example is the following (which can also, and equally, be parsed as a 2+):

[Soccer players] [in] [hospital?] (6,8)

[S] [O] [C]

Light = CENTRE FORWARDS (Times 4317)

Here the C-part generates a somewhat dubious (hence the polite question mark) synonym for hospital: centre for wards. If this is then split differently (if there's a shift in its word split), we can generate a synonym for Soccer players (CENTRE FORWARDS).

2.6.2 The palindrome

As palindromes are not numerous in English (though level and radar are common enough), CE has few to rely on for its C-parts. When they do occur, they almost invariably have special sub-operators such as whichever way you look at it, the same back and running back and forth. My current data set (Times 4305-4327) contains a single example:

> Boat unaffected by capsizing (5)
>
> Light = KAYAK (Times 4325)

In addition to this, there is one instance of a sub-type, as follows:

2.6.3 The mixed palindrome

In this variation, only part of the light is a palindrome. An instance is the word INTERPRET, where the final seven letters are palindromic. Hence the clue, with its ultra-literal sub-synonym:

> Explain meaning of popular palindrome (9) (Times 4320)

2.6.4 The double duty

In the double-duty clue-types, there is either no true light-synonym, or else a very weak one. Instead, an alternative reading of the C-part itself provides the primary-level synonymy. An example of a full double-duty clue is the case involving the light STYPTIC above (2.1.1) — something that you press to injured chin. An example of a part double-duty clue-type (where there is at least a weak light synonym) is:

> Possibly an MP, et al, I run into here (10)
>
> Light = PARLIAMENT (Times 4321)

The word here is an extremely weak synonym for PARLIAMENT since it can (as a deictic) stand for any place whatsoever. But more support for the light is given by the Standard English semantics of the clue which suggest a place where one might run into an MP. As we saw above in section 1 (General note on CE-SE relations), whole-clue semantics is ordinarily misleading if read in SE; but in this case we might say that, if it is misleading, that is only because it's abnormally helpful!

2.6.5 The light-clue reversal

Reversals are so rare that I can think of only one, now legendary, example (which I am no longer able to source):

> SGEG (9, 4)
>
> Light = SCRAMBLED EGGS (memory)

The form of the reversal is this: use an anagram sub-operator, scrambled, and actually make it part of the light! Hence clue and light are reversed: that is, if there were an English word SGEG, it could be given by the clue Scrambled eggs (4).

3. Conclusion

Whether the above is in fact a complete catalogue of ideal C-part- (hence overall clue-) types remains to be seen. Certainly it can't account for what might be called "the brilliant freak", a clue which deliberately and obviously breaks CE syntax and yet still works — perhaps because of insider knowledge, as in the following example:

> Crest ticketer? (7)
>
> Light = SPOONER (memory)

I can at least say, however, that the list covers every single case in my present data set. That data set, it must be remembered, is limited to instances from the Times — albeit that this puzzle is taken as a paragon of writing (compilation) in CE. How we are to cope with the various liberties that other compilers take with the basic grammar is, then, a moot point. For the more prescriptive amongst us, such liberties can amount to charges of poor usage, or even utter incompetence. For others, they will provide an opportunity to generate dialect and creole grammars. And finally, there is another pressing matter: even if we have been able to delimit the possible clue-types as ideal (though instanciable) forms, we have also seen that these forms are quite routinely hybridised in practice. This leads to a further research question and perhaps even a full diachronic investigation: are there rules for limiting hybridisation? Clearly there are some limits — so that, for example, anagrams of synonyms (so-called "concealed anagrams") are ungrammatical. On the other hand, there can be quite peculiar hybrids which flirt with ungrammaticality such as the combination of anagrams with eliminators:

> Florentine nearly crashed van (5,4)
>
> Light = FRONT LINE (Times 4327)

or eliminators with shift splits:

> City that was almost supreme (6)
>
> Light = THEBES (Times 4327)

But my suspicion beyond such general observations is that there are no fixed rules for hybridisation: this is simply the point at which we go beyond grammar into art.

Notes

*. E-mail <mchoul@murdoch.edu.au> Fax: +619 360 6367. I would like to thank Garry Gillard and Michael O'Toole for their useful comments on this draft.

1. The data set for the present paper is 23 puzzles numbered 4305 to 4327 in the Australian newspaper's reprint of (London) Times puzzles. (To translate the Australian puzzle numbers to their British originals, add a factor of 15,809.) This puzzle is taken as a model not only of correctness and parsimony by compilers and solvers alike, but also of the art of compilation. A few additional instances are from memory.

2. Although its parsing is open to ambiguity, it may be possible to read the following clue as CSO (an ultra-rare possibility):

> Bury bodies outside main church — crosses called for (10)
>
> Light = INTERSECTS (Times 4324)

Here, called for may be thought of as an operator (since it indicates the light synonym), or else crosses called for can be read as the light synonym in and of itself.

3. I would like to thank Richard Webb for coining this term.

4. We must write "almost but not quite" in this context because there is always the possibility of puns (2.3.3). In such cases, though, we can reasonably expect a question mark — an unlikely marker in a case of a genuine double synonym. A case in point is the following:

> Inspiring passages? (7)
>
> Light = BRONCHI (Times 4325)