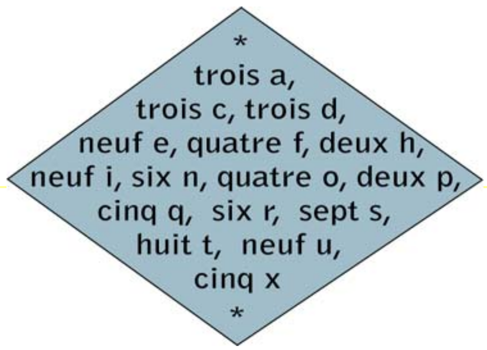
REFLEXICONS

by Lee Sallows

Reflexiconstruction

A lexicon is a dictionary or a list of words. Hence my use of "reflexive lexicon", or more crisply, *reflexicon*, for a self-descriptive word list that describes its own letter frequencies:



Immortal verity sans superfluity. Now that is what I call *belles lettres*. Below we shall look at some English examples. But first, since the answer is far from obvious, how are reflexicons produced?

Imagine a book with pages inscribed as follows. The text on page 1 might be anything - a colophon, an epigram, a dedication. For convenience we assume something short. But page 2 and all subsequent pages each comprise a descriptive list, in words, of the numbers of a's, b's, c's, etc., appearing on the previous page. Thus, if page 1 features a single *x* then our volume begins like this:

```
Page:  1    2      3       4          5          6        7 ..
       ------------------------------------------------------
       x  one x  one e  five e's   five e's      .        .
                 one n  five n's  three f's      .        .
                 one o  five o's  three i's      .        .
                 one x   one x      two n's      .        .
                                    two o's      .        .
                                  three s's      .        .
                                  three v's      .        .
                                    one x        .        .
                                                 .        .
```

This may not be the recipe for a bestseller but the plot does have its appeal. You can hardly help wondering how it ends. Will list lengths continue to expand? Clearly 26 items is the limit. In fact here none will exceed 16. These will be totals for E,F,G,H,I,L,N,O,R,S,T,U,V,W,X,Y, which are the only letters occurring in English number words under ONE HUNDRED, a number much higher than feasible list entries, assuming brevity in the opening text. Our example is therefore a *lipogram*, a work in which A,B,C,D,J,K,M,P,Q,Z will be absent because missing from page 1, the only page on which they could first occur. The end of our story can now be discerned.

Every new page shows a list of at most 16 totals, none of them large. The possible variations are thus finite. Sooner or later the numbers on one page will recur on another, albeit differently ordered. Suppose the totals on page *N* are the same as those on page *M*. Then page *N* is an *anagram* of page *M*; their letter frequencies agree. But this means that page *N*+1 will be identical to page *M*+1, which shows that our book must wind up in a repetitive cycle. And the same will be true whatever the starting text. Call the number of pages occurring in such a cycle its *period*. If the period is *P* then we have a closed loop of *P* sequentially descriptive lists. If *P* = 2 they will form a mutually descriptive pair. If *P* = 1 then we have a list whose description is a copy of itself: a reflexicon.

Let distinct letters stand for distinct lists. The onset of a period 1 loop, *R*, then looks like this: ..,*L*,*M*,*N*,*O*,*P*,*Q*,*R*,*R*,*R*,*R*,.. This shows that the reflexicon *R* not only describes itself, but it describes list *Q*, as well. So *Q* must be an anagram of *R*, most probably a different ordering of the same set of totals. Once any of its anagrams turn up, the reflexicon itself follows immediately. No reflexicon is reached *except* via one of its anagrams, unless of course we start off with the reflexicon itself on page 1.

Question: Assuming no A/B/C/D/J/K/M/P/Q/Z on page 1, how many different loops are there? Using a computer to extend the above shows that its pages converge on a loop of period 155. Extended trials reveal that provided we stick to priming texts using the 16 cardinal letters only (none to occur more than 99 times), there are just four possible outcomes. One is the loop of period 155, another is of period 14, while the remaining pair are both of period 1, the two basic English reflexicons:

fifteen e's,
seven f's, four g's,
six h's, eight i's,
four n's, five o's,
six r's, eighteen s's,
eight t's, four u's,
three v's, two w's,
three x's.

sixteen e's,
five f's, three g's,
six h's, nine i's,
five n's, four o's,
six r's, eighteen s's,
eight t's, three u's,
three v's, two w's,
four x's.

The two longer loops are readily reconstructed by extrapolating from any of their constituent lists, such as the following (condensed into digits):

E	F	G	H	I	L	N	O	R	S	T	U	V	W	X	Y	length
14	7	0	3	6	0	7	6	6	18	8	4	3	3	3	0	155
17	3	1	5	5	0	4	5	5	14	8	1	2	2	2	0	14

Thus, in English, all the multitude of different possible starting positions lead inexorably into one of these four whirlpools or *attractors* as mathematicians call them (see Doug Hofstadter's admirably lucid exposition in [2]). This convergence is easy to understand. A 16 element list has 16! = 20,922,789,890,000 distinct permutations (= anagrams), all of them giving rise to a common description which is itself one among 16! new lists having a common successor, and so on. The resultant funnelling effect carries interesting implications.

Consider a computer program able to generate pages for such a book, starting from any text. A basic routine scans TEXTIN = page *N*, initially page 1, counts its letters and writes their totals in the form of number-words to TEXTOUT = page *N*+1. TEXTOUT is now substituted for TEXTIN, the routine reiterated, and so on. I like to picture this process as a machine that takes text as input and yields text as output, the latter coupled back to the former via a feedback path. This makes it easier to see that a reflexicon is effectively a *virus*: a code sequence able to subvert the machine so as to get itself perpetually reproduced. One way to hunt for reflexicons is therefore to set such a machine going and just wait for contagion to set in. However, there are still other viruses that may easily usurp it first. These are the loops of longer period, all of them similarly infectious. How can we immunize the device against these unwanted invaders? How do we write a book that ends specifically in a loop of period 1?

One answer is to alter the mechanism so as to neutralize longer cycles. Instead of updating the totals for every letter on every page, suppose the next page results from correcting the total for a single letter chosen at random each time. The resulting haphazard behaviour is loop-free by definition except in one case: when updating a total entails no change in the subsequent list because it is already correct — because the list *is* already a self-descriptor. In this way the program is forever free to keep juggling numbers until it eventually succumbs to a self-reproducer. The only snag is that anagrams of a solution then pass unheeded, which means 16! chances lost every time. But not if we *alternate* methods: all totals updated on one pass, one random correction the next, and so on repeated. Now the former will catch any anagram, while the latter prevents latch-up in loops. A few million iterations (mutations) normally suffice to evolve (naturally select) a viable solution (virus). Assuming one exists, of course, failing which the process grinds on unchecked.

I should like to add that the key idea of loop-busting through inclusion of a *random* factor in the iteration process was the invention of John R. Letaw, a consultant in the areas of high-energy physics and astrophysics. Letaw had been the first to respond to a foolhardy challenge of mine that appeared in *Scientific American*, by coming up with an algorithm that yielded:

This computer-generated pangram contains six a's, one b, three c's, three d's, thirty-seven e's, six f's, three g's, nine h's, twelve i's, one j, one k, two l's, three m's, twenty-two n's, thirteen o's, three p's, one q, fourteen r's, twenty-nine s's, twenty-four t's, five u's, six v's, seven w's, four x's, five y's, and one z.

In fact, Letaw's algorithm worked quite differently to the one described above, in his scheme, successive approximations being determined by a weighted averaging process. Later, I experimented extensively with his algorithm, ending up with the method here outlined, which retains a random component although quite differently applied. Details of Letaw's algorithm can be found in [1], which appeared in response to an article of mine in [4].

Reflexiconography

Skipping refinements, so much for the basic machinery. What can we do with it? For a start, note that a self-descriptive *sentence* is really a sugar-coated reflexicon, the essential kernel overlaid with some palliative dummy text such as, "This sentence contains ..". Thus, on appending these constant ballast letters to successive counts, our standard process again issues in an associated self-descriptive list, provided it exists. If not, change "contains" to "employs", say, and try again. Passing over the simplest instances, a few special finds made after adapting the mechanism to suit the purpose deserve notice here. These are seen below in: *Example* 1, a (British) letter-totalling sentence, *Example* 2, an (American) letter-totalling self-descriptive pangram, and *Example* 3, a (trans-Atlantic) mutually-descriptive (pangrammatic) pair; cf. [3], and *Example* 4, a mutually-descriptive pair showing identical dummy text. Details of the program changes entailed by these special types would occupy us unduly, the basic mechanism remains the same.

> This sentence contains one
> hundred and ninety-seven letters:
> four a's, one b, three c's, five d's,
> thirty-four e's, seven f's, one g, six h's,
> twelve i's, three l's, twenty-six n's, ten o's,
> ten r's, twenty-nine s's, nineteen t's,
> six u's, seven v's, four w's,
> four x's, five y's, and one z.

Example 1

This pangram contains
two hundred nineteen letters:
five a's, one b, two c's, four d's,
thirty-one e's, eight f's, three g's, six h's,
fourteen i's, one j, one k, two l's, two m's,
twenty-six n's, seventeen o's, two p's,
one q, ten r's, twenty-nine s's, twenty-
four t's, six u's, five y's, nine w's,
four x's, five y's, and one z.

Example 2

The right-hand
sentence contains
four a's, one b, three c's,
three d's, thirty-nine e's,
ten f's, one g, eight h's,
eight i's, one j, one k,
four l's, one m, twenty-three n's,
fifteen o's, one p, one q,
nine r's, twenty-three s's,
twenty-one t's, four u's,
seven v's, six w's,
two x's, five y's, and one z.

The left-hand
sentence contains
four a's, one b, three c's,
three d's, thirty-five e's,
seven f's, four g's, eleven h's,
eleven i's, one j, one k,
one l, one m, twenty-six n's,
fifteen o's, one p, one q,
ten r's, twenty-three s's,
twenty-two t's, four u's,
three v's, five w's,
two x's, five y's, and one z.

Example 3

The adjacent text utilizes
four a's, one b, two c's,
three d's, thirty-six e's, five f's,
three g's, nine h's, eleven i's,
two j's, one k, four l's, one m,
eighteen n's, thirteen o's,
one p, one q, eight r's,
twenty-seven s's, twenty-four t's,
four u's, four v's, seven w's,
three x's, four y's, and two z's.

The adjacent text utilizes
four a's, one b, two c's,
three d's, thirty-two e's, nine f's,
three g's, eight h's, eleven i's,
two j's, one k, three l's, one m,
seventeen n's, fifteen o's,
one p, one q, eleven r's,
twenty-six s's, twenty-one t's,
eight u's, six v's, six w's,
three x's, four y's, and two z's.

Example 4

Returning to reflexicons proper, in line with French practice above, the plural S is dispensible. Two instances are then found, one trivial :

FIVE F, FIVE I, FIVE V, FIVE E.

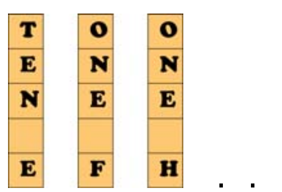
the other less dull:

TWELVE E,	FIVE R,
SIX F,	FIVE S,
THREE H,	SIX T,
SEVEN I,	THREE U,
TWO L,	SIX V,
TWO N,	FOUR W,
FIVE O,	FOUR X.

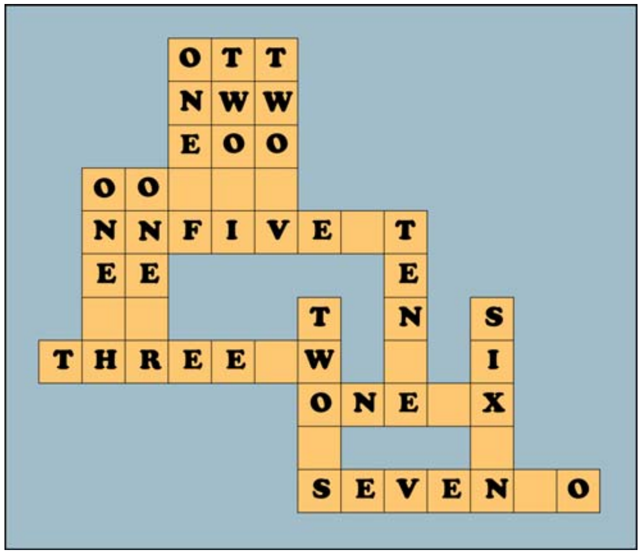
This is condensed, but logologists like their alphabet soup really thick. Plural S has been dropped. Is there any other way to increase the semantic density through discarding still further redundant symbols? There is.

Consider a list in which the stated letter counts are in each case exactly one short of the true total: `TEN E, ONE F, ONE H, TWO I, SIX N, SEVEN O, ONE R, TWO S, FIVE T, TWO V, THREE W, ONE X.` That is, there are *eleven* e's, not ten, *two* f's and not one, etc. Each of the twelve items on the list can now be written on a strip of card, on one side running from left to right:

on the other from top to bottom:



Using trial and error, an arrangement must now be sought such that the strips overlap eachother in a self-descriptive crossword pattern that eliminates excess letters. The following shows my own very first attempt at a *self-intersecting* reflexicon:

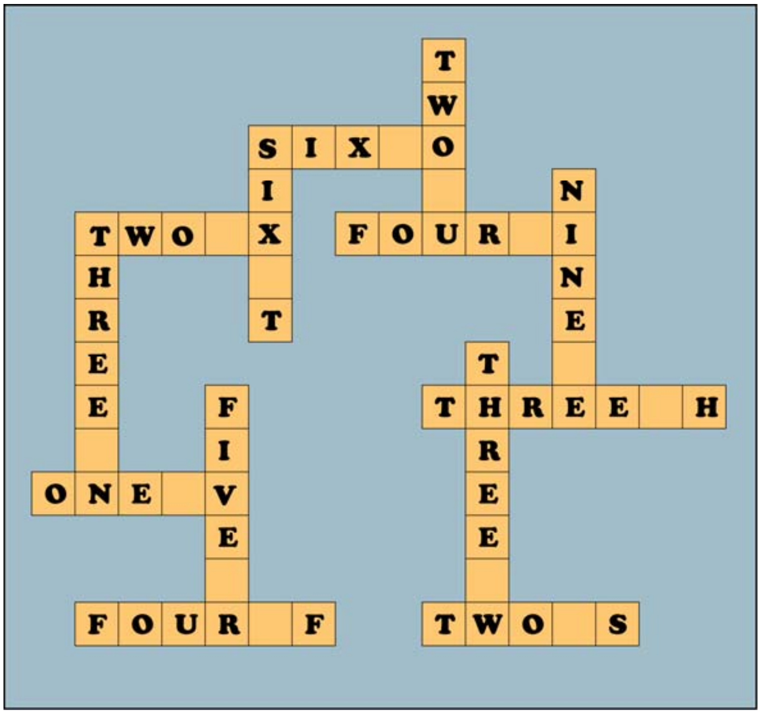


Example 5

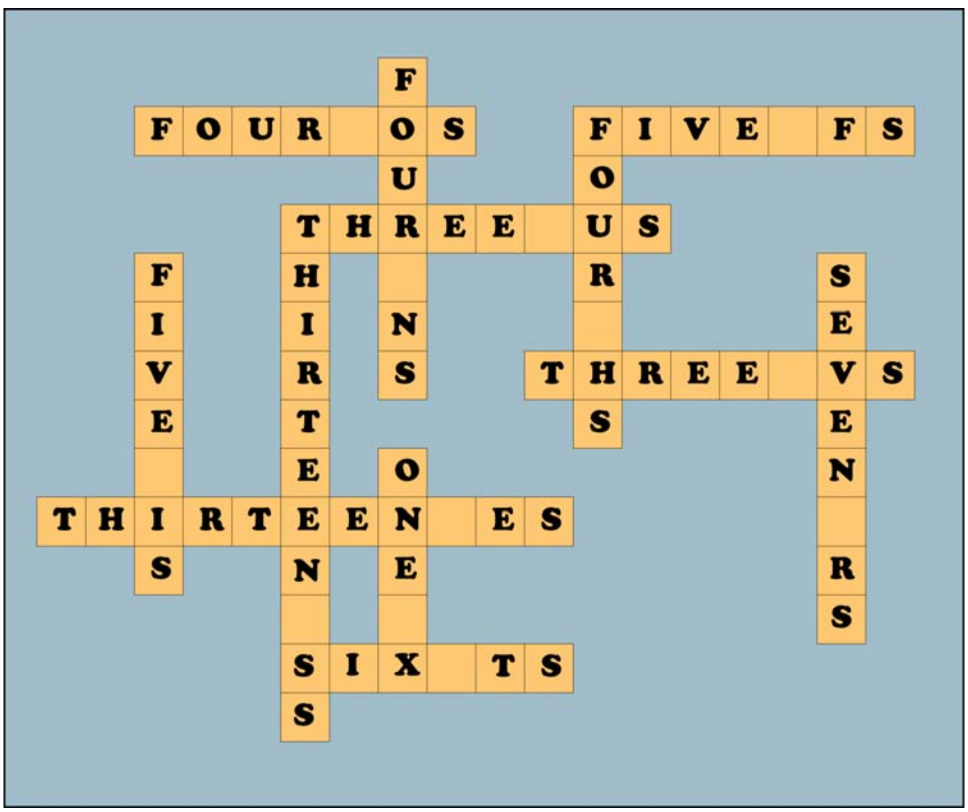
This was a good start, but OTT, NWW, EOO, OO, NNFIVE, and EE, are pseudowords and thus serious blemishes. A different layout wont help either since ONE H and ONE R must always remain bonded together with HR in THREE. No, to escape this problem called for a new set of items involving less intersections per strip so as to win elbow-room. This brings us to a key insight.

Twelve strips bearing 12 excess letters imply 12 intersections. Yet N strips can cross at most N-1 times unless linked to include a closed chain. Look at ONE X, SIX N, SEVEN O and TWO S in Example 5. Contriving such a loop is the major constraint in devizing solution layouts. Thus, a new list requiring fewer intersections than strips makes for a big gain in layout flexibility (and vice versa), although two or more fewer will imply a non-connected pattern. To avoid this, the obvious course then is to seek an N item list involving N-1 excess letters = intersections. An example is seen in *Example 6*.

This is more like it: no pseudowords and 3.846 letters per word or 0.26 words per letter, which, with the words now spatially interlocking, is virtually alphabet jelly! The trouble is that now one letter (F) is alone in not occupying an intersection, a niggling asymmetry. At some loss in semantic density, however, restoring plural S is another way to win room for maneuver, as in *Example 7*.



Example 6

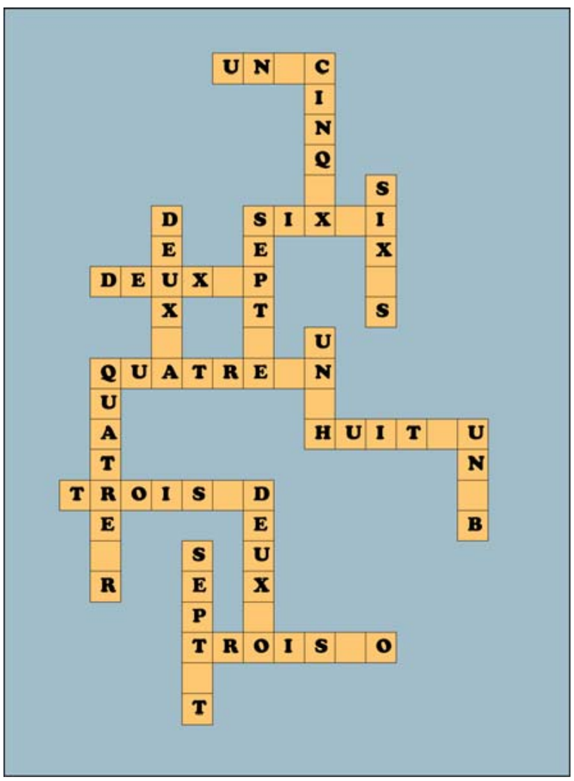


Example 7

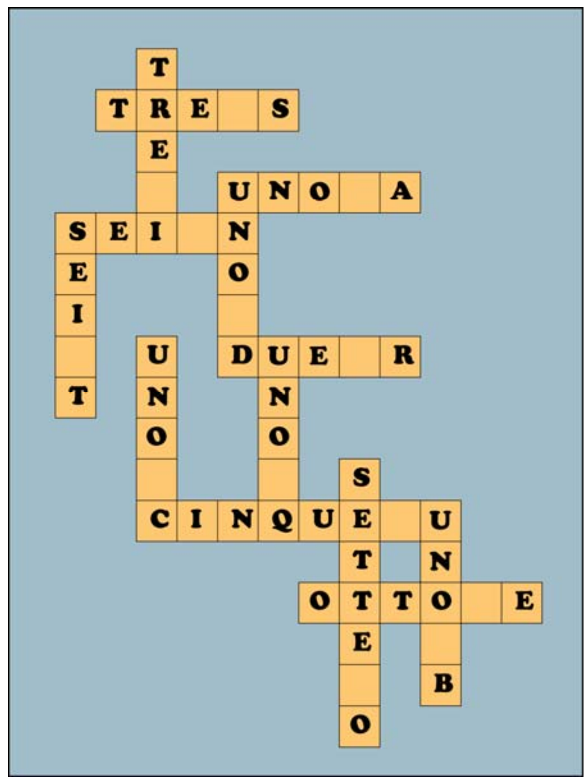
Here we are back to 12 strips and 12 intersections (necessitating a loop), each occupied by one of the 12 letters occurring. On consideration, this is a remarkable property, more so than first sight suggests, since it depends on finding a list in which the letters outnumber their totals by one exactly, the excess then vanishing on intersects. The list used in *Example 7* is thus exceptional. For example, no French or Italian equivalent exists. Unusually, however, English enjoys two such lists, the second comprising 13 words, although its internal peculiarities impede the construction of elegant self-intersecting layouts. Some readers may like to try their hand; the totals are as follows: E:15, F:8, G:1, H:3, I:5, L:1, N:4, O:5, R:5, S:11, T:4, U:3, V:4.

Of course, there is nothing against letters appearing on intersects more or less than once, as with U (2/3 times) and B/A&B (not at all) in the French and Italian Examples 8 and 9, neither of which languages call for plural S.

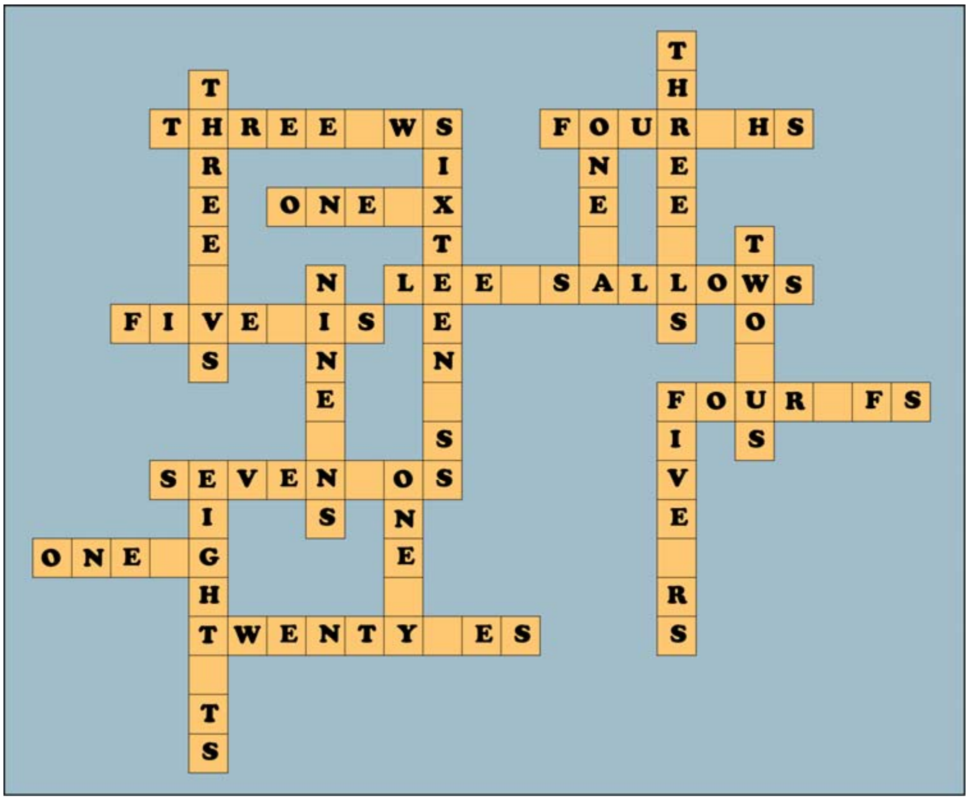
Example 8



Example 9

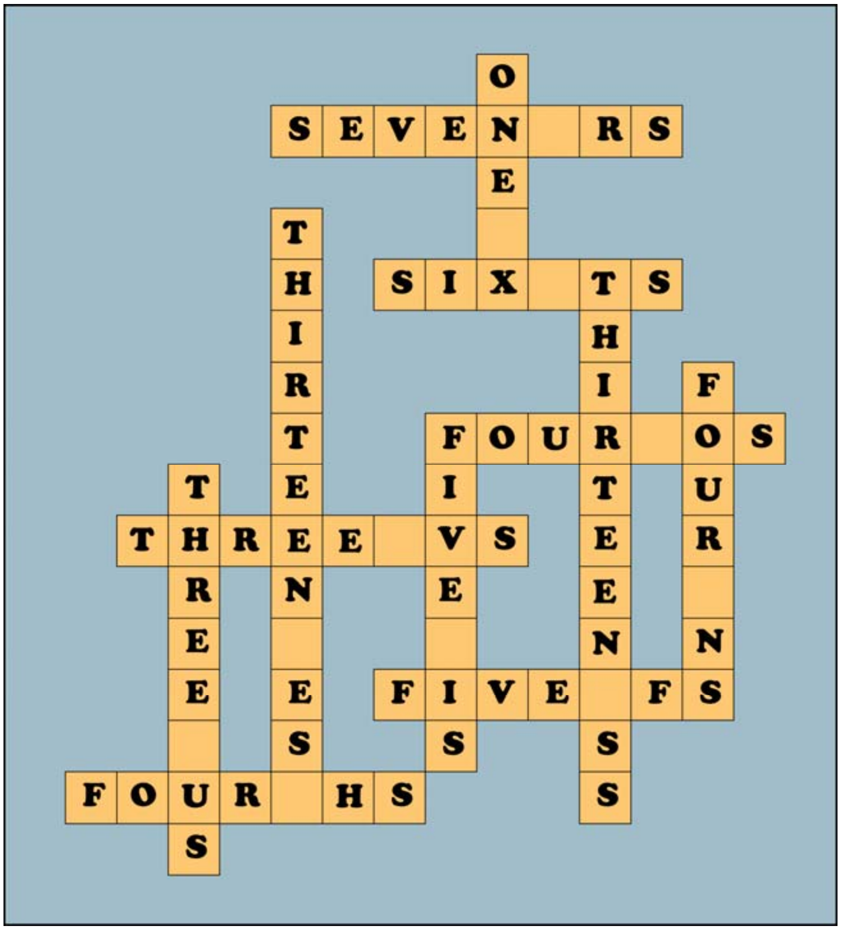


Both of these illustrate a further trick in the reflexiconographer's repertoire: the use of "ONE #" (here "UN B" and "UNO A/B") as unobtrusively appended dummy text. This is a useful stratagem when "pure" solutions cannot otherwise be found, although the arbitrariness of letter used (UN B could equally be UN Z) detracts from their logological elegance, a point to bear in mind when assessing the merits of different specimens. Dummy text may take more conventional forms of course, as in *Example 10*, where intersects outnumber strips, a fact reflected in multiple loops. However, the construction of such specialities is demanding, to say the least.



Example 10

Some loops are not what they seem. *Example 11* exhibits pseudoloops and the two ways they arise: via intersection on a blank; viz. THIRTEEN SS and FIVE FS, and via abutment onto a blank, viz. THIRTEEN ES and FOUR HS. The single real loop here is formed by FOUR OS, FOUR NS, FIVE FS, and FIVE IS.

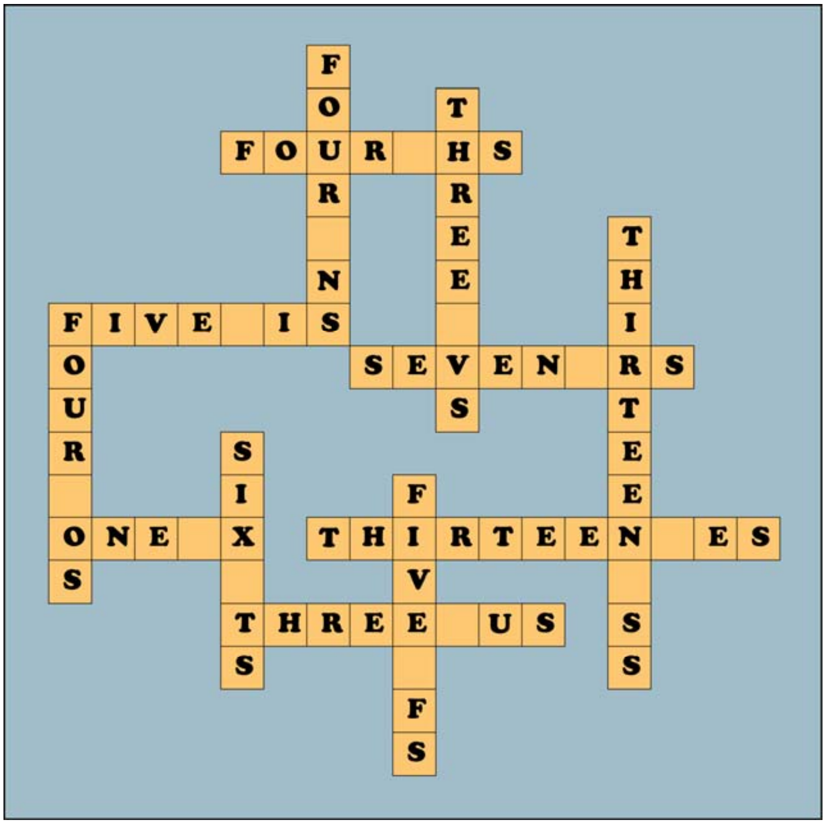


Example 11

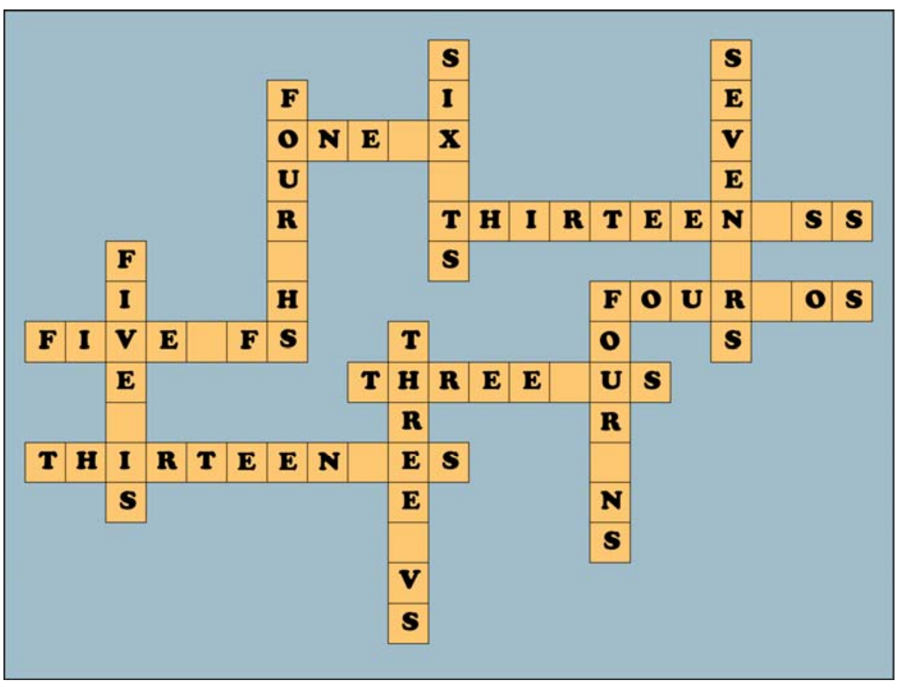
Pseudoloops can make for compacter layouts, a fact seen in comparing *Examples 11* and *7* both of which, be it noted, use the same entries (the special set of 12 strips), whereas the two patterns occupy rectangles of 14 × 16 versus 14 × 18, respectively. Two natural questions then arising are: How many distinct (fully connected) self-intersecting reflexicons can be formed from this set of strips? and, Which of them is the most compact?

To seek answers, Victor Eijkhout, a mathematical friend, wrote a recursive strip-shuffling computer program able to scan for solutions. However, although several days running on a mainframe computer produced thousands of alternative solution layouts, it became clear there was no chance of the job terminating within any feasible time-scale. The two questions thus remain unanswered. *Example 11,* which was hand-produced, is the most compact specimen known.

Nevertheless, at my suggestion Victor set his (slightly modified) program to work on a new but related search that was to bear fantastic fruit. *Examples 12* and *13* embody two jewels of logology (we seem to have reached alphabet ice). Here are the classic strips again, the loop now realized as the entire set holding hands in a single twelve-linked bracelet! The pair shown are among 18 such specimens found by the program, not counting rotations and reflections, but including trivial variations such as when FIVE IS is switched with FIVE FS in Example 12.

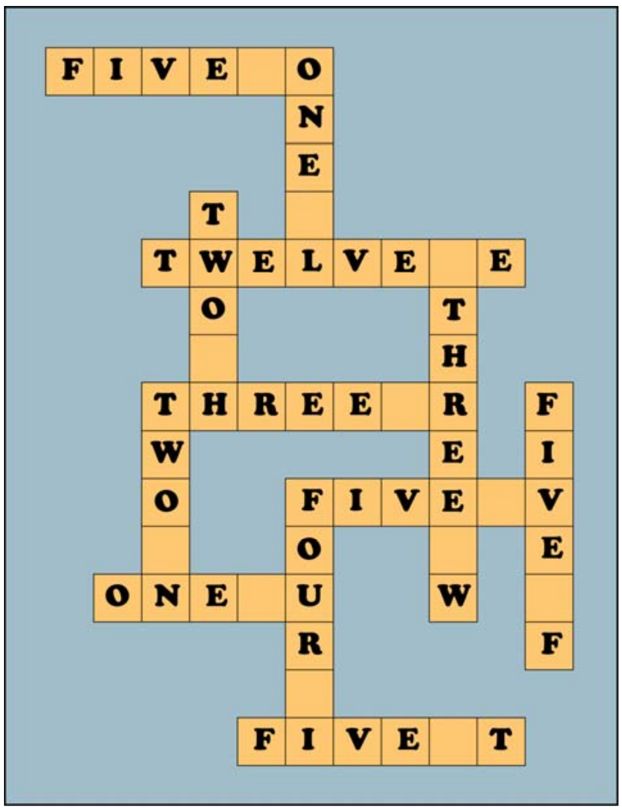


Example 12



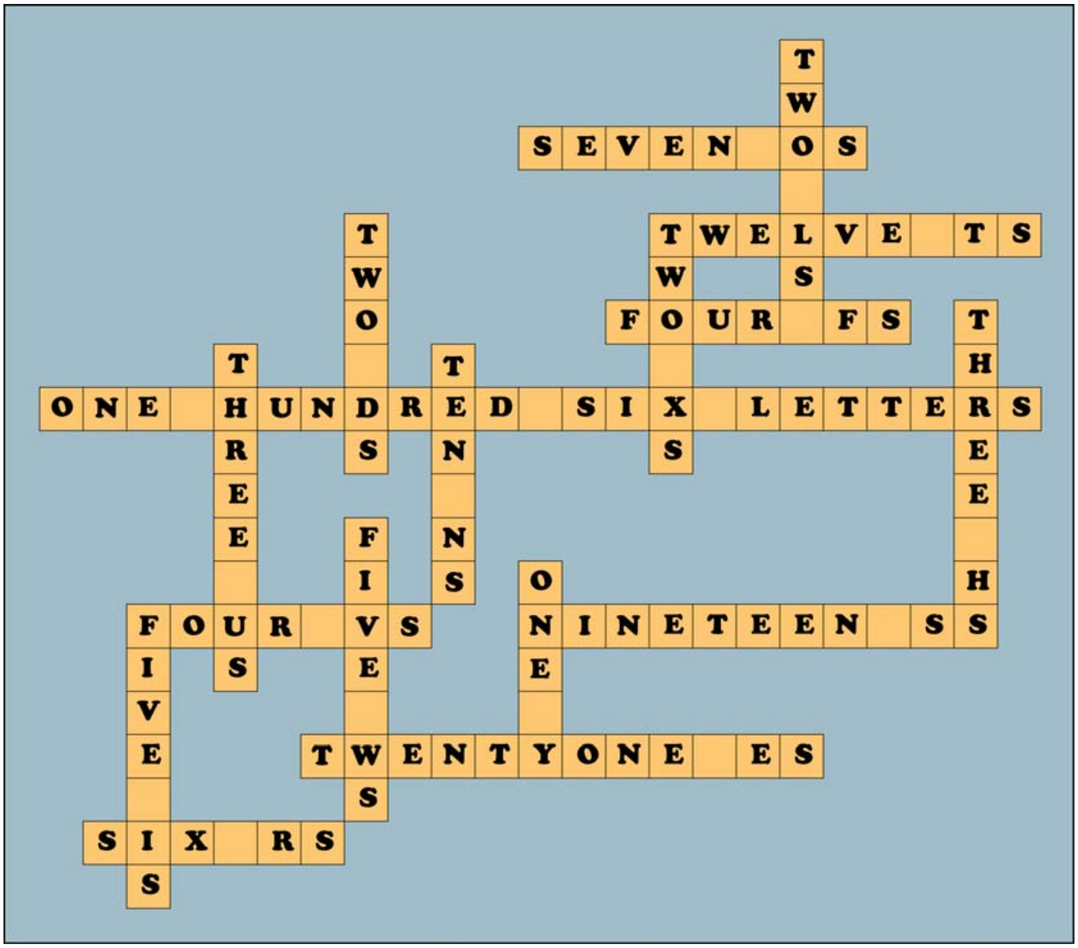
Example 13

Marvellous as Eijkhout's finds are, further collector's pieces probably await discovery. For example, might there exist a reflexicon with a truly symmetrical layout? A congruent pair showing distinct solution entries? A 3-dimensional bracelet (that forms a knot)? A (possibly interlacing) co-descriptive pair? A pangrammatic reflexicon (without dummy text)? The list is easily extended. In the meantime one special specimen has passed unmentioned. *Example 14* again features 12 intersections each occupied by one of the 12 letters occurring, although now there is no plural S. It is a relative of *Example 5,* the first self-intersector examined, where the number of excess letters also matches the number of items, but which cannot be solved without creating “pseudowords". As with the list used in *Example 7*, a second list with the same property (but minus plural S) has been found. (A third trivial case is FOUR [F], FOUR [O], FOUR [U], FOUR [R]). The analogous question then arises: How many distinct solutions can be formed from the entries in *Example 14*?



Example 14

Lastly, to conclude this brief review, in *Example 15* I offer a final example of the state of the art, a reflexicon that incorporates its own letter-total. Can a similar specimen be found using still fewer than 106 letters? Another tough challenge for the computational logologist! However, the seemingly insuperable problem that keeps me awake at nights is how to produce a self-descriptor that will tell us what letters it uses where?



Example 15

* * * * * *

The above article was first published in *Word Ways*, August 1992

<u>References</u>

[1] Letaw John R., Abacus, Vol. 2, No.3, pp 42-7
[2] Hofstadter D. Metamagical Themas, Basic Books, 1985. p.364
[3] Brooke Maxey, Pangrams, Word Ways Vol.20 No.2, p.90
[4] Sallows L., In Quest of a Pangram, Abacus, Vol.2, No.3, pp 22-40.