In Quest of a Pangram

by Lee Sallows

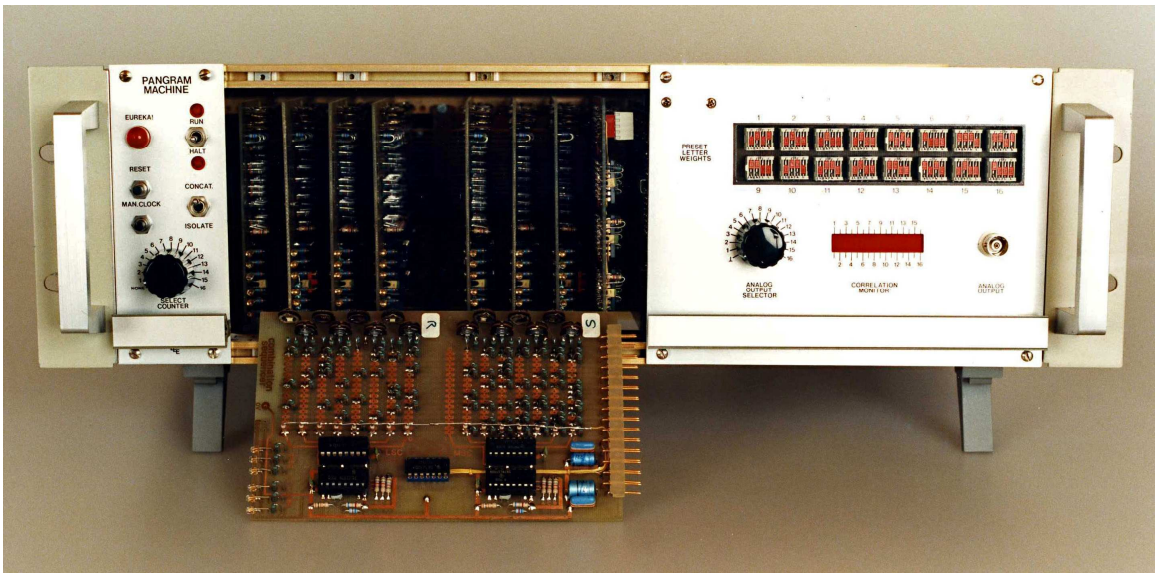
The Pangram Problem

In February 1983, a Dutch newspaper, the *NRC Handelsblad*, carried an astonishing translation of a rather tongue-in-cheek sentence of mine that had previously appeared in one of Douglas Hofstadter's *Scientific American* columns ("Metamagical Themas," January 1982). Both the translation and an article describing its genesis were by Rudy Kousbroek, a well-known writer and journalist in Holland. Here is the original sentence:

Only the fool would take trouble to verify that his sentence was composed of ten a's, three b's, four c's, four d's, forty-six e's, sixteen f's, four g's, thirteen h's, fifteeen i's, two k's, nine l's, four m's, twenty-five n's, twenty-four o's, five p's, sixteeen r's, forty-one s's, thirty-seven t's, ten u's, eight v's, eight w's, four x's, eleven y's, twenty-seven commas, twenty-three apostrophes, seven hyphens and, last but not least, a single *!*

Complete verificiation is a tedious task; unsceptical readers may like to take my word for it that the number of letters and signs used in the sentence do indeed correspond with the listed totals. A text which inventories its own typography in this fashion is an example of what I call an *autogram* (*autos* = self, *gramma* = letter). Strict definition is unnecessary, different conventions giving rise to variant forms; it is the use of cardinal number-words written out in full that is the essential feature. Below we shall be looking at some in which the self-enumeration restricts itself to the letters employed and ignores the punctuation.

Composing autograms can be an exacting task, to say the least. The process has points in common with playing a diabolically conceived game of patience. How does one begin? My approach is to decide first what the sentence is going to say and then make a flying guess at the number of occurrences of each sign. Writing out this provisional version, the real totals can be counted up and the initial guess updated into an improved estimate. The process is repeated, trial and error leading to successively closer approximations. This opening soon shades into the middle game. By now all of the putative totals ought to have been corrected to within two or three of the true sums. There are, say 9 f's in fact but only *seven* being claimed, and 27 real t's where *twenty-nine* are declared. Switching *seven* with the *nine* in *twenty-nine* to produce *nine* f's and *twenty-seven* t's corrects both totals at a single stroke. Introducing further cautious changes among the number-words with a view to bringing off this sort of mutual cancellation of errors should eventually carry one through to the final phase.



One of the eight printed circuit cards bearing two counters and the resistor fan-outs for letters R and S. At left, two groups of three LEDs signal current counter positions.

The end game is reached when the number of discrepancies has been brought down to about four or less. The goal is in sight but, as in a maze, proximity is an unreliable guide. Suppose, for instance, a few day's painstaking labour have at last yielded a near-perfect specimen: only the x's are wrong. Instead of the *five* claimed, in reality there are 6. Writing *six* in place of *five* will not merely invalidate the totals for e, f, s, and v, the x in *six* means that their number has now become 7. Yet replacing *six* with *seven* will only return the total to 6. What now?

Paradoxical situations of this kind are a commonplace of autogram construction. Interlocking feedback loops magnify tiny displacements into far-reaching upheavals: harmless truths cannot be stated without disconfirming themselves. Clearly, the only hope of dehydrating this Hydra and getting every snake-head to eat its own tail lies in doctoring the text accompanying the listed items. In looking at the above case, for example, only a fool will fail to spot istances where style has been compromised in deference to arithmetic. Short of a miracle, it is only the flexibility granted through choice of alternative forms of expression which would seem to offer any chance of escape from such a labyrinth of mirrors.

This is what made Kousbroek's translation of my sentence so stunning. Number-words excepted, his rendering not only adhered closely to the original in meaning, it was simultaneously an autogram in Dutch!

Or at least, so it appeared at first sight. Counting up, I was amused to find that three of the sums quoted in his sentence did not in fact tally with the real totals. So I wrote to the author pointing out these discrepancies. This resulted a month later in a second article in the same newspaper. Kousbroek wrote of his surprise and dismay on being caught out by the author of the original sentence, "specially come over from America, it seems, to put

me right." The disparities I'd pointed to, however, were nothing new to him. A single flaw had been spotted in the supposedly finished translation on the very morning of submitting his manuscipt. But a happy flash revealed a way to rectify the error in the nick of time. Later a more careful check revealed that this "brainwave" had in fact introduced even more errors elsewhere. He'd been awaiting "the dreaded letter with its merciless arithmetic" ever since. The account went on to tell of his titanic struggle in getting the translation straight. The new version was included; it is a spectacular achievment.

The tail concealed a subtle sting, however. At the end of his story Kousbroek threw out a new (letter-only) autogram of his own:

Dit pangram bevat vijf a's, twee b's, twee c's, drie d's, zesenveertig e's, vijf f's, vier g's, twee h's, vijftien i's, vier j's, een k, twee l's, twee m's, zeventien n's, een o, twee p's, een q, zeven r's, vierentwintig s's, zestien i's, een u, elf v's, acht w's, een x, een y en zes z's.

A finer specimen of logological elegance is scarcely conceivable. The sentence is written in flawless Dutch and couldn't possibly be expressed in a crisper or more natural form. In ordinary translation it says, "This pangram contains five a's, one b, two c's, … (etc) … one y, and six z's." [A *pangram*, I should explain, is simply a phrase or sentence containing every letter of the alphabet at least once (*pan* = all, *gramma* = letter). In the present article we are looking at self-enumerating pangrams, or pangrams which are simultaneously autograms. In such pangrams, some letters will occur only at the point where they themselves are listed (look at k, o, q, u, x, y, above]. Following this pangram came a devilish quip in my direction: "Lee Sallows will doubtless find little difficulty in producing a magic English translation of this sentence," wrote Kousbroek.

Needless to say, I didn't manage to find any errors in *this* sentence of his.

Autograms by Computer

Rudy's playful taunt came along at a time when I had already been looking into the possibility of computer-aided autogram construction. Anyone who has tried his hand at composition will know the drudgery of keeping careful track of letter totals. One small undetected slip in counting can later result in days of wasted work. At first I had envisaged no more than an aid to hand composition: a program that would count letters and provide continuous feed-back on the results of keyboard-mediated surgery performed on a sentence displayed on screen. Later I began to wonder what would happen with a program that cycled through the list of number-words, checking each against its corresponding real total and making automatic replacements where necessary. Could autograms be evolved through a repetitive process of selection and mutation? Several such LISP programs were in fact written and tested; the results were not unpredictable. In every case processing would soon become trapped in an endless loop of repeated exchanges. Increasing refinements in the criteria to be satisfied before a number-word was replaced would win only temporary respite from these vicious circles.

What seemed to be needed was a program that could look ahead to examine the ramificat-ions of replacing *nineteen* by *twenty*, say, before actually doing so. But how is such a program to evakuate or rank prospective substitutions? Goal-directed problem solving converges on a solution by using differences between intermediate results and the final objective so as to steer processing in the direction of minimizing them. The reflexive character of autograms frustrates this approach. As we have seen, proximity is a false index. "Near-perfect' solutions may be anything but near in terms of the number of changes needed to correct them, while a sentence with as many as eight discrepant totals might be perfected through replacing a single number-word. If hand-composiiton is obliged to rely on a mixture of guesswork, word-chopping, prayer, and luck, how can a more intelligent strategy be incorporated into a program?

I was pondering this impasse when Kousbroek's challenge presented itself, distracted my attention, and sent me off on a different tack. The sheer hopelessness of the undertaking caught my imagination. But was it actually impossible? What a comeback if it could really be pulled off! The task was to complete a letter-only autogram beginning, "This pangram contains … ." A solution, were it discoverable, must in a sense exist "out there" in the abstract realm of logological space. It was like seeking a number that has to satisfy certain predetermined mathematical conditions. And nobody — least of all Kousbroek — knew whether it existed or not. The thought of finding it was a tantalizing possibility. Reckless of long odds, I put aside programs and launched into a resolute attempt to discover it by hand-trial.

It was a foolhardy quest, a search for a needle in a haystack without even the reassurance of knowing that a needle had been concealed there in the first place. Two week's intermittent effort won only the consolation prize of a near-perfect solution: all totals correct save one; there were 21 t's instead of the 29 claimed. With a small fudge, it could even be brought to a shaky sort of resolution:

ttttt
t
t
this pangram contains five a's, one b, two c's, two d's, twenty-seven e's, six f's, three g's, five h's, eleven i's, one j, one k, two l's, two m's, twenty n's, fourteen o's, two p's, one q, six r's, twenty-eight s's, twenty-nine t's, three u's, six v's, ten w's, four x's, five y's, and one z.

To the purist in me, that single imperfection was a hideous fracture in an otherwise flawless crystal. Luckily, however, a promising new idea now suggested itself. The totals in the near-solution must represent a pretty realistic approach to what they would be in the perfect solution, assuming it existed. Why not use it as the basis for a systematic *computer search* through neighbouring combinations of number-words? Each of the near-solution totals could be seen seen a centered in a short range of consecutive possibilities within which the perfect total was likely to fall. The number of f's, say, would probably turn out to lie somewhere between two and ten, a band of nine candidates clustered about

"six". With these ranges defined, a program could be written to generate and test every combination of twenty-six number-words constructible by taking one from each. The test would consist in comparing these sets of potential totals with the computed letter frequencies they gave rise to, until an exact match was found. Or until all cases had been examined. Blind search-ing might succeed where cunning was defeated.

PROFILES

It isn't actually necessary to deal with all twenty-six totals. In English there are just ten letters of the alphabet which never occur in any number-word between zero and hundred, the one too low and the other too mhigh to appear in the pangram. These are a, b, c, d, j, k, m, p, q, and z. The totals for these letters can thus be determined from the initial text and filled indirectly:

This pangram contains five a's, one b, two c's, two d's, ? e's, ? f's, ? g's, ? h's, ? i's, one j, one k, ? l's, two m's, ? n's, ? o's, two p's, one q, ? r's, ? s's, ? t's, ? u's, ? v's, ? w's ? x's, ? y's, and one z.

This leaves exactly sixteen critical totals. Counting up shows that there are already 7 e's, 2 f's, 2 g's, 2 h's, 4 i's, 1 l, 10 n's, 11 o's, 2 r's, 24 s's, 7 t's, 1 u, 2 v's, 5 w's, 1 x, and 1 y: sixteen constants which must be added to those letters occurring in the trial list of sixteen number-words.

Though straightforward in principle, the program I now set out to write carried its practical complications. Number-words lack the regularity of numerals (in whatever base notation), still less the harmony of the numbers both stand for. An obvious step was to replace number-words by PROFILES: alphabetically ordered sixteen-element lists representing their letter content. The PROFILE for *twenty-seven*, for instance, would be:

e	f	g	h	i	l	n	o	r	s	t	u	v	w	x	y
3	0	0	0	0	0	2	0	0	1	2	0	1	1	0	1

The letters above the list are for guidance only, and form no part of the PROFILE itself. A special case was the PROFILE for one, which provided for the disappearance of plural s ("one x, two x's") by including –1 in the *s* position. PROFILES for all number-words up to *fifty* (anything higher than *forty* was unlikely ever to be needed) were stored in memory, and a label associated with each. These labels were chosen to coincide with the number represented. The label for the PROFILE for *twenty-seven*, for example, would be the decimal number 27.

Starting with the lowest, a simple algorithm could now generate successive combinations of labels (that is, numbers) drawn from the sixteen pre-defined ranges. We shall return to these in a moment. Each set of labels would be used to call up the associated set of PROFILEs. These sixteen PROFILEs would be added together element for element, and the resulting sums in turn added to the above-mentioned constants so as to form a SUMPROFILE; see Figure 1. The SUMPROFILE would thus contain the true letter

frequencies for the presently activated sentence (the sixteen number-words represented by the current combination of labels plus residual text). All that remained was for the program to check whether the numbers in the SUMPROFILE coincided with the present set of PROFILE labels. If so, the candidate combination of number-words agreed with the real totals and the pangram had been found. If not, generate the next combination and try again… .

The SUMPROFILE

LABEL	PROFILE																NUMBER-WORD	LETTER
	e	f	g	h	i	l	n	o	r	s	t	u	v	w	x	y		
27	3	0	0	0	0	0	2	0	0	1	2	0	1	1	0	1	twenty-seven	E
6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	six	F
3	2	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	three	G
5	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	five	H
11	3	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	eleven	I
2	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	two	L
20	1	0	0	0	0	0	1	0	0	0	2	0	0	1	0	1	twenty	N
14	2	1	0	0	0	0	1	1	1	0	1	1	0	0	0	0	fourteen	O
6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	six	R
28	2	0	1	1	1	0	1	0	0	0	3	0	0	1	0	1	twenty-eight	S
29	2	0	0	0	1	0	3	0	0	0	2	0	0	1	0	1	twenty-nine	T
3	2	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	three	U
6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	six	V
10	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	ten	W
4	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	four	X
5	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	five	Y
	7	2	2	2	4	1	10	11	2	24	7	1	2	5	1	1	INITIAL TEXT CONSTANTS	
	27	6	3	5	11	2	20	14	6	28	21	3	6	10	4	5	SUMPROFILE	

Figure 1

A stack of PROFILEs and initial text constants are added to produce a SUMPROFILE. The example shown is the hand-produaced near-perfect pangram. All SUMPROFILE and label numbers coincide except that for T.

The simplicity of this design conveys no hint of the uncounted alternatives reconnoitered before reaching it. The “obvious” PROFILEs were not quite so conspicuous as suggested, being in fact a later improvement over a previous look-up table. Weeks were spent in exploring a quite different approach which sought to exploit the mutual-cancelling technique formerly used in hand-composition. By the time the final version of the program had come into focus, half a dozen prototypes lay behind and several months had slipped by. In the meantime, cheeerful enthusiasm had given way to single-minded intensity as the problem wormed its way under my skin. Neither was I working entirely alone. Word of the pangram puzzle had spread among colleagues, discussion sprang up, and contending design philosophies were urged. At one stage, complaint of “excessive CPU-time devoted to word games” came in from the University of Nijmegen Computing

Centre, whose facilities had beeen shamelessly pressed ito service. This was when rival programs were running simultaneously. It was bad enough to be in search of a Holy Grail that might not even exist; the thought of someone else finding it first added a sticky sense of urgency to the hunt.

The question of determining the exact ranges of number-words to be examined seemed to me an essentially trivial one, and I put it off until last. The important thing was to get the program running. For the time being it was enough to decide what the lowest combination was going to be, and to let the algorithm generate all possibilities up to, say, ten higher for each number-word. In terms of software it was convenient for ranges to be of equal length; ten might be unnecessarily high, but better the net be too large than that the fish should escape. Since the totals in the near-solution were to define the midpoint of these ranges, their lower limits would commence at about five less. "Fourteen o's," for instance implied a range running from nine up to eighteen (or perhaps ten up to nineteen). The values actually settled upon — on the basis of pencil and paper trials with near-autograms — can be seen in Figure 2. Ranges for each of the sixteen critical letters are represented as vertical scales with numbers (standing for number-words) indicating their starting and finishing totals. Within these ranges fall the hand-produced near-solution sums tracing out a histogram silhouette. In most cases these are, by definition, situated roughly in the middle of the range. For the low totals, *l*, *g*, and *u*, however, this is impossible: in a pangram all letters must occur at least once; the range cannot extend below one (see Figure 2).

Combinatorial Explosion

At long last the program was finished and set going. Roughly a million combinations had already been tested during the development period. The trouble with previous versions had been their hopelessly slow speed. Even the latest program could only test something like ten new combinations per second. This was still sluggish, but bearing in mind the hefty letter-crunching involved (16×16 additions in calculating the SUMPROFILE alone, for example), I thought it probably couldn't be greatly improved upon. Vaguely I wondered how long it would take before a solution popped up. Being a greedy consumer of valuable processor time, the program ran at nights as a low-prioity "batch-job" on the Computing Centre's VAX 11/780 machine. Every morning I would hasten to call up the job file, running my eye swiftly down the screen in search of "EUREKA!", which would precede a printed record of the magic combination of number-words. As day succeeded day without result, the question of how long it would be before all possibilities had been exhausted gradually assumed importance. It was a matter I had never given any serious thought. 10^7 cases had already been examined. Let's see, how many would there be altogether … ?

The calculation is an absurdly simple one and even now I blush to recall what the result implied. Programatically the ten totals in each of the sixteen ranges are cycled exactly like the 0 – 9 digits on the rotating number-discs of the familiar tape-counter or odometer. Advancing this software counter a single step results in the next combination of totals

being clicked into position, ready for the pangram test. The all-zero state will correspond to the first or lowest set of number-words: the bottom row of scale numbers in Figure 2. Just as the mechanical counter begins at 0 and steps in turn through every number (that is, through every possible digit sequence) up to the highest, so the program runs through all possible combinations up to that coinciding with the top row in Figure 2. In effect, we are systematically examining every single histogram that can be plotted. About halfway through the process, the example shown for the near-solution totals will come up for testing. How many such graphs can be drawn in Figure 2? The answer is clearly the same as that number displayed on our sixteen-digit odometer after stepping through all possible positions: a string of sixteen 9s (plus one for the zero position) = 10^{16}. Is there a golden vein running through the ten-deep strata? A milky nipple crowning the Gaussian breast? At a speed of ten combinations per second, to find out is going to take $10^{16}/10$ seconds. A pocket calculator soons converts this to more intelligible units. There seemed to be something wrong with the one I was using. Every time I worked it out the answer was ridiculous: 31.7 *million years!*

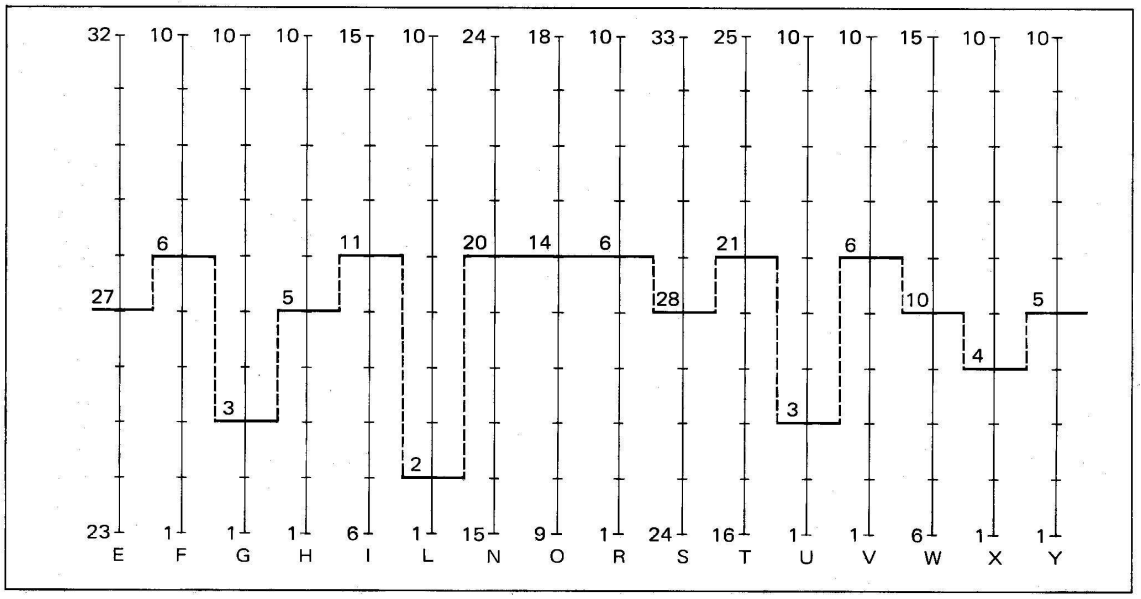


Figure 2

The range of frequency values to be considered for each letter that appears in number-words.

I was so unprepared for the blow contained in this revelation that initially I could hardly take it in. The whole object of turning to a computer in the first place had been to canvass huge numbers of combinations fast. Now that the truth had dawned, I began cursing my naivete in ever embarking on such a fool's errand. True, I was an electronics engineer, not a professional programmer. However, the more I contemplated the kind of speeds at which a realistic program would have to run, the more preposterous the whole computer venture appeared. Conceivably a somewhat faster program could be written. But even checking at a rate of one million combinations/second, it would take three hundred and seventeen years to run through the ten-deep range of possibilities.

Yet thoughts of millions of combinations per second put me in mind of *megahertz*. And megahertz brought my mind back to Electronics. This in turn prompted an idea, a fanciful notion, for the first few days no more than an idle phrase repeated in the head, a good title perhaps for a science fiction story: *The Pangram Machine*.

Initially I didn't take the thought seriously. I was disconsolate after the embarrassing failure of the computer project, and the absurd expression "pangram machine" mocked hollowly at the back of conciousness. Yet suddenly the vague intuition began to crystallize; in a flash I saw how a central process in the program could be simulated electronically. Taking this mechanism as a starting point, I tried translating other aspects of the algorithm into hardware. It worked; it was easy. A few hours later, I was amazed and thrilled to find the broad outlines of an actual design already clear in my mind.

The Phoenix now emerging from the ashes of the Pangram Quest soared serenly to the sky, smoothly circled, swiftly swooped, and soon bore me off, a helpless prisoner in its relentless talons. For the next three months I would be pouring all my energy into the construction of a high-speed electronic Pangram Machine.

The Pangram Machine

How seriously should a word puzzle be taken? Though only the size of a smallish suitcase, the apparatus to emerge from three months of intense activity packed more than two thousand components onto thirteen specially designed printed circuit cards. More than a hundred of these were integrated circuits or "chips", each containing on the average something like fifty transistors. Foresight of this conplexity might have disuaded me from starting. In the event, the completed machine turned out to involve a good deal more electronics than originally planned. Readers uninterested in technical details may prefer to skim the following section.

At the heart of the device is the electronic equivalent of a continuously-stepped sixteen-digit odometer: a clock-driven cascade of sixteen Johnson-counters; see Figure 3 for all that follows. The clock is a simple 1 MHz square-wave generator producing a continuous train of 10^6 pulses every second. As mentioned above, however, even checking at this rate, ten-deep ranges would take 317 years to explore. A reduction was therefore demanded, the choice of new range-length being primarily determined by the availability of standard 8-output devices. Each counter is thus a circuit having 8 outputs, which become consecutively activated by successive pulses presented to its single input. Before the clock is started, a RESET button on the control panel (see photo, page ?) enables all counters to be initialized or "zeroed", meaning that all "0" outputs are made active. As the clock ticks, the activated output of the first counter in the chain changes from "0" to "1" to "2", etc., so that after seven clock pulses output "7" will be activated, whereupon the next pulse reactivates "0" and the process begins anew.

Coupling between counters is like that between odometer discs in that, after completing one cycle, it is arranged for a single pulse to be sent to the input of the following counter

in the cascade. Eight cycles of the first are thus needed to step the second counter through one. In this way every new clock pulse results in activating a unique combination of sixteen output lines. Afet 8^{16} pulses, all combinations will have been run through and, unless halted, the entire process will begin again.

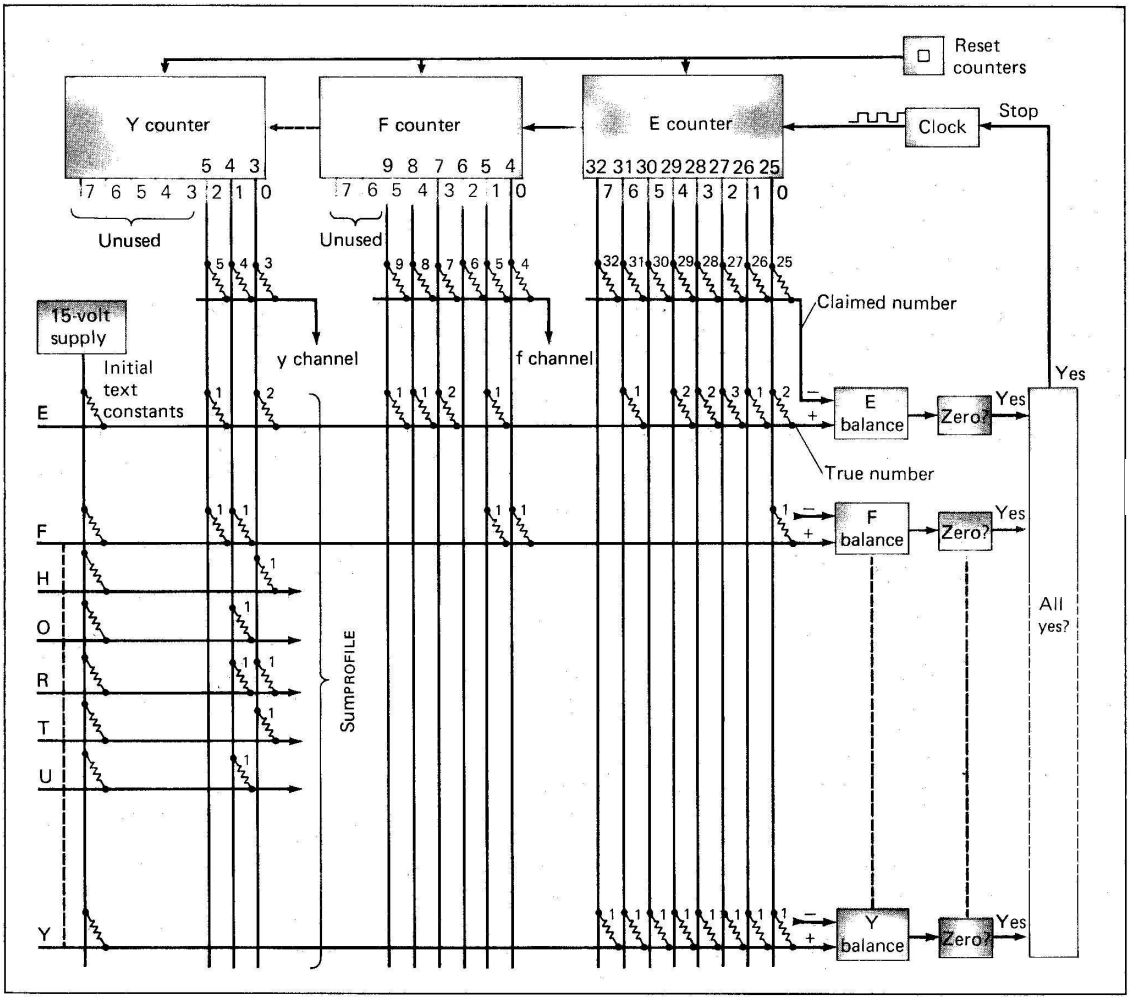


Figure 3

The design of the Pangram Machine.

Even so, calculation shows that running time must still be measured in *years* unless some further limitations are introduced. In fact, the cycle-length of counters is individually presettable. With a preset cycle-length of 5 for instance, a counter's "0" line becomes reactivated on the sixth input pulse, while outputs "5", "6", and "7" remain unused. In this way, the range-length for different letters is individually adjustable, and a shorter total running time can be achieved (at the price of narrower ranges). Figure 3 shows that the y-counter's cycle-length has been reduced to 3, for example. Later we shall turn our attention to the actual set of ranges used.

Now just as in the computer program, the object of activating different combinations of output lines is to call up sets of PROFILEs whose corresponding elements will be added together so as to form a SUMPROFILE (as discussed above; I leave the initial text constants temporarily out of account). Electronically the instantiation and addition of PROFILEs can be achieved using either digital or analog techniques. The former is far preferable, but costly. The analog technique is less predictable in performance but, in this case at least, made attractive by its relative simplicity. Here, as elsewhere, financial limitations meant that design was influenced by what the junk-box had to offer. In the end, I was forced to use an analog approach; but since other parts of circuitry are digital (the counters, for instance) the overall design is really a hybrid.

Accordingly, the PROFILEs "called up" by activated counter outputs take the form of resistor fan-outs feeding specific patterns or profiles of discrete current levels into sixteen common lines representing the SUMPROFILE. Every counter output is associated with a pre-determined number-word (shown in counter boxes). An activated output is one transistor-connected to a 15 volt supply and thus able to deliver current; nonactivated outputs are simply left unconnected (these are so-called open-collector outputs). The PROFILE of each number-word is implemented as a set of resistors connecting the counter output to appropriate SUMPROFILE lines. These are the horizontal lines E, F,… (H…O…R…T, U,) …Y shown in the diagram. (Sixteen 0.5 ohm resistors, not shown but electrically important, connect each of these to ground or zero volts).

Current drawn from the activated outputs thus divides into a number of resistor-adjusted streams and is distributed over the E, F, … Y lines of the SUMPROFILE so as to represent the contribution of each PROFILE-number. PROFILE summing is thereby achieved almost without doing anything: the current produced in each SUMPROFILE line (and hence the voltage over its 0.5 ohm resistor) is simply the aggregate of the sub-currents injected into it via the resistors in the currently activated set of PROFILEs.

The number and value of the resistors used in each case depends entirely on the PROFILE being simulated. Choosing an arbitrary unit of current to represent one letter, double this value will stand for two, and so on. In fact, with the exception of *seventeen*, which alone contains four e's, values in the PROFILEs are always 0, 1, 2, or 3. Since 0 is indicated by no current = no connection, all PROFILEs (excepting that for *seventeen*) can be implemented by resistor sets built up from just three discrete values of resistance: x ohms, x/2 ohms, and x/3 ohms, yielding current levels of 1, 2, and 3 units, respectively. (In reality x = 3920 ohms, a high value relative to the 0.5 ohm resistor over which the sum voltage falls; this is important for achieving summing linearity). A concrete example is shown for the *y*-counter's *three* and *four*. The small diagonal zigzags are the resistors. The numbers printed alongside represent not their resistance, but the number of current units (15 volts / 3920 ohms = 3.82 milliamps) they pass into the SUMPROFILE line: *three* = 2 e's, 1 h, 1 r, 1 t; *four* = 1 f, 1 o, 1 r, 1 u.

So far so good: the current entering each ' + ' input of the boxes marked BALANCE is a measure of the number of e's, f's, etc., actually occurring in the present set of sixteen activated number-words; every microsecond a new set is switched in. But the

SUMPROFILE is incomplete without the initial text constants — themselves comprising no more than a special PROFILE and thus representable as a set of fixed-bias currents. Hence a further array of sixteen resistors permanently connected from the 15 volt supply to each SUMPROFILE line (see Figure 3).

Now in the program SUMPROFILE, totals (representing true letter frequencies) are compared with the labels of the PROFILEs (the numbers corresponding to the number-words) to check for complete agreement. These label-numbers are simulated by an extra resistor-determined current derived from each counter output (top rows of resistors). E-label currents are fed to the ' – ' input of the E-BALANCE box, F-label currents to the ' – ' input of the F-BALANCE box, and so on. Comparison of SUMPROFILE and label currents takes place in the BALANCE boxes; each box is a differential amplifier whose output voltage is a fixed multiple (the amplification factor) of the difference between its two input currents (or voltages, depending on how you look at it). In this way SUMPROFILE and label-numbers are weighed against each other in the BALANCE; only if they are equal will the output voltage be zero or close to zero volts. Of course, all sixteen pairs are weighed simultaneously.

The rest ought to be obvious. The "ZERO?" boxes are window-detectors; circuits signalling a logical 1 ("yes") if their input voltage lies within a predetermined voltage range or "window". The window in this case is a narrow one centered on zero volts (+/- 50 mV). All window-detector outputs go to a sixteen-input AND-gate ("all yes?"). If sixteen zeroes turn up together, the AND-gate will fire, stopping the clock, freezing the counters, and turning on an inessential but comforting EUREKA! lamp mounted on the control panel. The magic set of number-words sought will now be represented by the frozen combination of activated outputs. In order to signal which these are, counter positions are indicated (in binary code) in the form of sixteen groups of three light-emitting diodes (LEDs) visible through a red plexiglass front panel. Using a table to translate LED patterns into number-words, it will remain only to double-check the result by hand and, if it is correct, ring for the champagne.

Though all very well on paper, in reality the analog techniques used in the machine are messy. Circuit capacitance and amplifier settling times set a practical limit to speed of operation. When the clock ticks and switches in a new set of PROFILEs, electronic havoc breaks loose as overshoots, oscillations, glitches and gremlins conspire to drive window-detectors into palsied indecision. After a while, electrons begin to simmer down and circuits settle out into a new steady state. For this reason, rather than going straight to the STOP input of the clock as shown in Figure 3, the AND-gate output is actually sampled some 900 nanoseconds after the clock pulse onset — that is, at the last moment of the clock cycle, only 100 nanoseconds before the next pulse arrives. This idea, among others, was due to Willie van Schaijk, without whose friendly and expert assistance the machine might never have left the ground. Using the (TTL) technology at my disposal, a clock frequency of 1 MHz is the highest I was able to achieve under these circumstances. Given more funds, it would probably not be difficult to improve on this by a factor of ten. Digital techniques bring their own problems; I am not convinced that a worthwhile gain in speed could be won for the large investment needed.

Although all sixteen counters have eight outputs each, it is impossible to exploit these unrestrictedly, since to examine all possible combinations at a clock rate of 1 MHz would still take $8^{16}/10^6$ seconds = 8.9 years. Range lengths were therefore tailored to each letter so as to retain a reasonable chance of finding the pangram while bringing the running time down to about one month. Flexibility was maintined by providing printed circuit cards with easily alterable solder-links allowing preadjustment of each counter's cycle length. Selection of the ranges to be used was a ticklish business, involving careful analysis of letter frequencies in number-words. Those finally settled upon can be seen in Figure 4 (numbers under RANGE stand for number-words).

Notice that e, having a high frequency and being therefore less predictable than other letters, receives the maximum range length of 8. On the other hand, y, occurring exactly once in every number-word from *twenty* upwards, but in no others, can appear only 3, 4, or 5 times in the pangram *given the ranges* for e, n, s, and t. This is hardly a trivial insight: were y's range-length increaed to 4, ten days would be added to running time. As it is, to run through the combinations generated by the ranges in Figure 4 will take $(8 \times 6 \times 6 \times 6 \times 7 \times 4 \times 7 \times 6 \times 6 \times 7 \times 7 \times 6 \times 6 \times 7 \times 6 \times 3)/10^6 = 31.36$ days. Anything longer would have been unendurable.

In the program, the PROFILE for *one* contained –1 in the s-position to cancel what would otherwise be an s too many in the initial s-constant. However, minus values are not resistor-representable in the machine. As seen in Figure 4, there are only three letters (l, u, x) in whose ranges *one* occurs. To deal with these cases, after reducing the initial s-constant by 3, an s is added to the PROFILEs of number-words higher than *one* in their ranges. The range for l thus becomes: *one, two + s, three + s, four + s*; in other words, number-words above *one* bring their plural s with them. There is no reason why this couldn't be done for every number-word in every range (with corresponding reduction in the s-constant), but it would mean a lot of extra resistors.

Failure

After twelve weeks concentrated effort, the machine drew near to completion. As a prototype, it had posed a host of technical problems to be faced and overcome. First there had been a pilot phase to investigate the feasibility of an analog implementation. How fast could the critical summing and balance circuitry perform? Despite normal pessimistic expectations, small-scale trials yielded promising results. The only way to discover whether the full-scale version would function satisfactorily was to build it. At length the long program of design and construction culminated on the day the machine stood ready for a critical test: would it successfully identify and halt at a magic combination?

To find out, I introduced deliberate changes in the resistor-represented initial text constants; by feeding the machine with false data about letter frequencies in the introductory text, I could trick it into halting at a prearranged pseudo-magic combination.

Subtracting o and adding an i and n should cause it to stop at that combination of real totals represented in the previously discussed hand-produced solution: “twenty-one,”, the true number of t’s, then replacing “twenty-nine”. Using the “manual clock” and “select counter” controls to preadvance the five highest or “most-significant” counters in the odometer chain (u, v, w, x, y) to their appropriate totals (3, 6, 10, 4, 5), it would take only a few minutes for the faster-cycling counters to reach the remaining numbers in the magic combination. Starting the clock, I watched anxiously as the changing pattern of binary-coded LED displays reported the steady increment of counter positions.

Ranges of Number-Words

LETTER	NEAR-SOLUTION TOTAL	RANGE	RANGE LENGTH	INITIAL CONSTANT
E	27	25–32	8	7
F	6	4–9	6	2
G	3	2–7	6	2
H	5	3–8	6	2
I	11	8–14	7	4
L	2	1–4	4	1
N	20	17–23	7	10
O	14	12–17	6	11
R	6	3–8	6	2
S	28	24–30	7	21
T	21	18–24	7	7
U	3	1–6	6	1
V	6	3–8	6	2
W	10	7–13	7	5
X	4	1–6	6	1
Y	5	3–5	3	1

Figure 4

Suddenly and soundlessly the counters locked, the EUREKA! lamp came on, and the correlation monitor confirmed sixteen hits in a row. This was it; the machine had passed the acid test. With the correct text constants loaded and a few other loose ends tied up, one week later all was ready for the launching of this singular rocket on its thirty-two day voyage into the unexplored regions of logological space.

Lift-off came on 3 October 1983, almost eight months following the publication of Rudy Kousbroek’s audacious challenge. Cees Wegman, a spiritual godfather to the project who had watched sympathetically through the long months as I gracelessly declined from suave insouciance to crazed intensity, came along to perform the deed of honour. A bottle of wine was broached, and three of us sat with glasses raised as he ceremoniously clicked the starting switch to RUN (it was a fitting tableau for some quixotic latter-day Velsaquez, I couldn’t help musing).

The ensuing period found me hovering nervously over the machine. Among other things, there was the nagging worry of machine reliability: what guarantee was there of faultless operation over so long a period? The answer of course was *none*. All I could do was maintain sporadic surveillance with an oscilloscope, and halt the machine at three-day intervals to perform checks with the psuedo-magic combination. After a while the suspense became nerve-racking. Mornings were worst. On waking, the first thought in conciousness would be: *has it halted*? It took nerves of iron to go through the morning’s ablutions before tensely decending to the living room where the machine was installed on my writing bureau. Opening the door with great deliberation, I would quickly go in and

tranfix the machine with a questioning gaze. And there would be the flickering LEDs as the counters slowly switched their way through the 2.71×10^{12} combinations. One million a second for 31.366 days. It was a torturing experience. The novelty of watching the machine soon wore off and the edge of expectation blunted, but a single second's distracted attention was accompanied by the thought that another million chances had already elapsed, so perhaps NOW??? … and my glance would be wrenched back to the twinkling array of lights. After months of frenzied activity in building the machine, this period of enforced waiting was a cruel contrast of frustrated inertia and protracted disappointment.

But it was highly conducive to thinking up means for shortening that time. Before long, I saw that by halting the machine at key points in its travel and limiting the cycle-length of certain counters through calculated intervals, redundant checks on predictably invalid blocks of combinations could be obviated. Temporarily truncating the t-counter's range to exclude *eighteen* and *nineteen*, for instance, meant that all values of t contained a y so that y could occur only four or five times. Testing cases for which y = three could thus be skipped during such a phase. Using dodges of this kind, I was able to slice nearly ten days off the originally estimated running time.

Meanwhile the grains of sand —and of hope— were inexorably running out. Day succeeeded day with no sign of EUREKA! By 25 October, twenty-two days after launching, the machine had checked out every (undisqualified) combination of number-words within its capacity without finding the magic pangram. Since oscilloscope monitoring and a subsequent test with the modified initial text constants showed the machine to be functioning properly, I was not in any serious doubt about this negative result.

The crushing truth was that there never had been a needle in the haystack; the Quest for the Pangram had failed.

Logological Space

Though a bitter disappointment, the failure of the quest was not yet an irreversible defeat. A remote chance lingered that the magic combination lay yet undetected just outside the ranges of number-words examined. More promisingly, alternative translations remained to be explored. At the top of the list was "This pangram comprises …", a rendering of the Dutch *bevat* on a par with "contains".This would only entail a new set of initial text constants.

The prospect of another month in purgatory, however, was anything but inviting. Yet much had happened during the long weeks of waiting. In the range-limiting stratagem used to shorten the previous run had lain the seed of a powerful new development. Many hours' thought had been given to this, and already detailed preparations were in hand for a Mark II version of the machine incorporating extensive modifications.

Consider the number-words in the range for y: *three, four, five*; the letter y itself occurs in none of them. Put differently, whichever of y's PROFILEs may be activated, the actual number of y's can never be affected; in this sense, y is an independent variable. Great advantage can be taken of this by adding new circuitry which *measures* the number of y's present in the currently activated combination and uses the result to switch-in the appropriate y-PROFILE. In short, the y-counter can be replaced by an *automatic number-word selector*. And discarding the y-counter from the cascade will mean *dividing running time by three* (see Figure 5).

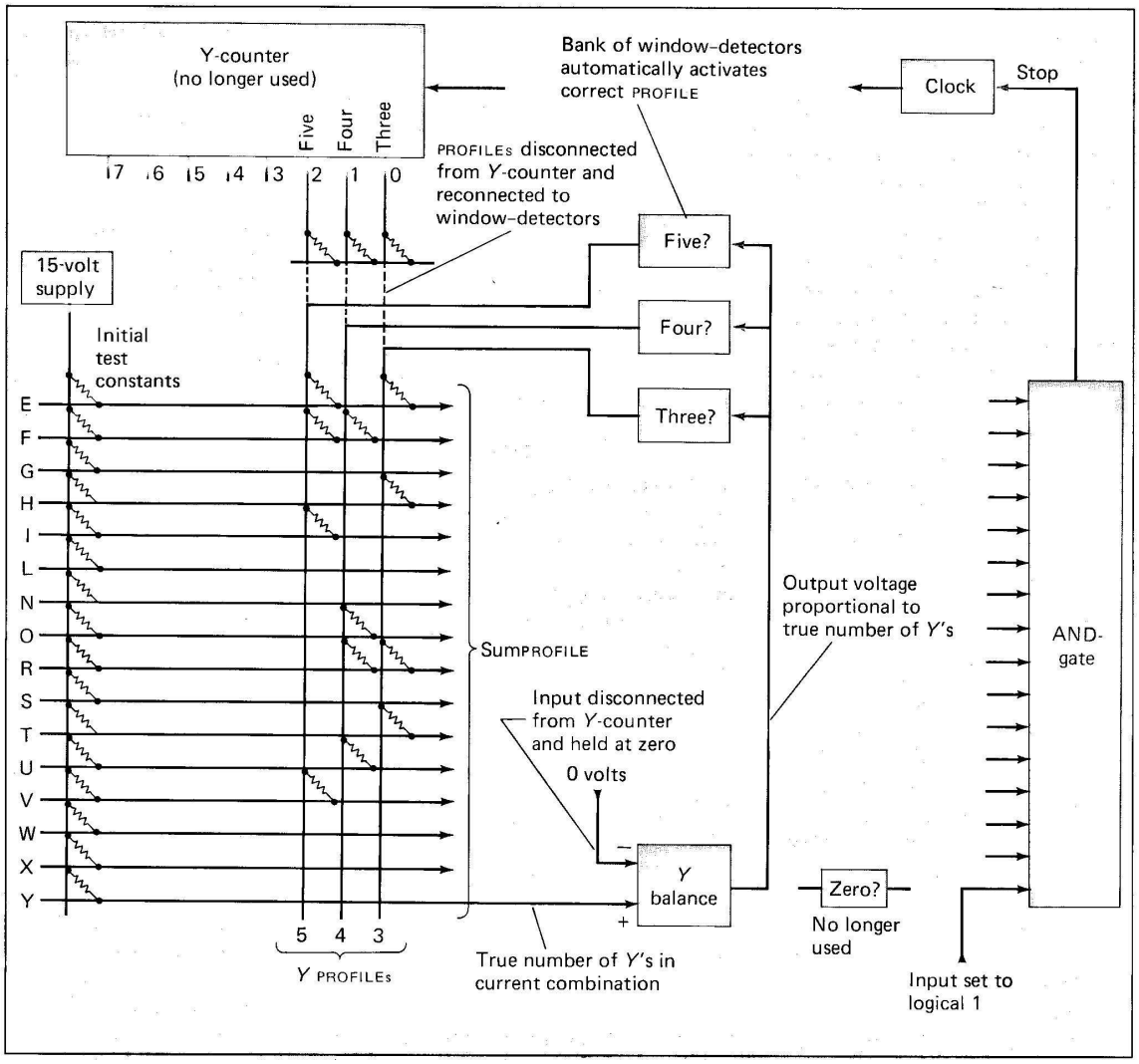


Figure 5

Example of automatic number-word selection applied to the letter y. A voltage proportional to the number of y's occurring in the present combination is classified by a bank of three window-detectors, one of whose outputs will activate the appropriate PROFILE.

The real power of this refinement emerges on seeing that the same trick can be worked for any letter not appearing in the number-words making up its own range. G and l are two such; providing *six* is dropped from its range, so is x. This then was the scheme to be realized in the blueprint for the new Mark II machine. With the g, l, x, y counters

removed from the cascade, running time falls to only $(8 \times 6 \times 1 \times 6 \times 7 \times 1 \times 7 \times 6 \times 6 \times 7 \times 7 \times 6 \times 6 \times 7 \times 1 \times 1)/10^6$ seconds or *one hundred and five minutes.* The perspective opened up by this dramatic improvement carried further implications in its wake.

With the ability to explore so quickly, means would be required for easy loading of different initial text constants. Though electrically trivial, a flexible resistance-selection method was difficult to implement in the machine as it stood. The final (and not altogether satisfactory) system chosen uses a set of four tiny switches for each channel. The latter work in binary fashion, so that a constant or "weight" of anything from 0 through 15 letters can be introduced. Incorporating this bank of 16×4 PRESET LETTER WEIGHTS switches on the front panel (see photo, page ?) involved some major surgery to the machine.

Another benefit of ultra-fast logological space travel is the chance to prospect further afield: that is, to expand ranges. Even if all twelve remaining counters are allocated a range length of 8 (the maximum available in this machine), running time comes out to only $8^{12}/10^6$ seconds = 19,08 hours. In two cases, l and y, the ranges of auto-selected letters may themselves be increased, an expansion that has its uses with initial texts containing l's and y's; for instance, "This pangram employs …". The g in *eight* and x in *six* make further extension impossible for g and x. In reality, impatience to get on dissuaded me from expanding range lengths until later so that running time was kept below two hours during initial explorations.

Besides serious mechanical alterations, the modifications sketched above called for a further printed circuit card carrying twenty-four new integrated circuits, the same number of transistors, and a few dozen associated components. The increased electrical drain meant in turn an extra dc power supply. Space was cramped, and the rise in internal heat dissipation threatened to upset the temperature-sensitive differential amplifiers. Notwithstanding these demands and difficulties, within a month the new souped-up Pangram Machine Mark II stood poised for its maiden flight.

Following a last-minute test with the modified initial text constants, now easy to enter via the front-panel switches, I started off with a re-check of "This pangram contains …". With running time down to under two hours, one could afford to be thorough. This time there was no wine, no ceremony, no Velasquez and, as anticipated, no result.

In the meantime I'd worked out the intial text constants for "This pangram *comprises* …", and as soon as the first run was over, I loaded these and set the machine searching again. Two hours later, the counter LEDs showed that the second run had been completed, and I was confronting a second disappointment. That truly was a tragedy; it meant that no really perfect English translation of the Dutch pangram existed. It seemed to me an unwarranted injustice, and, brushing aside a tear, I marked it down as another of the things I mean to ask God about on Judgment Day.

Even so, many excellent alternative renderings remained to be tried. These might not qualify as literal translations of *bevat* but would at least preserve the spirit of the original.

"This pangram *comprises* …" was therefore followed in quick succession by "This pangram *consists of*," "*is composed of*," "*uses*," "*employs*," and "*has.*" Every one of them without success!

By now I was beginning to wonder just how long this might go on. Given a random introductory text of, say, twenty-five letters, what is the probability that an associated self-enumerating list exists? Short of examining all possible twenty-five letter strings one at a time, I saw no way of answering the question. One in a hundred? One in a million? As it happens, the answer turns out to be something closer to one in ten.

On the second day of exploration I was sitting in front of the machine during its eighth run when suddenly the EUREKA! lamp came on and my stomach turned a somersault. Rigid with excitement, I carefully decoded the the LED dispalys into the set of number-words represented. A painstaking check completely verified the following perfect pangram:

This pangram lists four a's, one b, one c, two d's, twenty-nine e', eight f's, three g's, five h's, eleven i's, one j, one k, three l's, two m's, twenty-two n's, fifteen o's, two p's, one q, seven r's, twenty-six s's, nineteen t's, four u's, five v's, nine w's, two x's, four y's, and one z.

I leave it to readers to imagine the scenes of wild intemperance following upon this victory. Despite a hangover, next morning copies of the pangram were happily handed out among friends and colleagues who had patiently borne with me through the long months of pangrammania. Notable, if unsurprising, was that nobody felt disposed to examine the sentence for a discrepancy. Not unnaturally, I came in for a few words of congratulation, and some even looked at me with an unspoken "How does it feel to climb Everest?" on their lips. Like a dish-rag, actually; I still hadn't recovered from the previous evening's celebrations.

The zenith of glory was yet to come. Returning home at lunchtime, I found a magnificent trophy awaiting. I had set the machine running once more, early in the morning, and it had halted again at a new solution. Changing "and" to "&" in the natural English rendering of Rudy Kousbroek's pangram, a last desperate bid for a perfect magic translation had finally met with success. The Quest for the Pangram had ended in triumph!

This pangram contains four a's, one b, two c's, one d, thirty e's, six f's, five g's, seven h's, eleveni's, one j, one k, two l's, two m's, eighteen n's, fifteen o's, two p's, one q, five r's, twenty-seven s's, eighteen t's, two u's, seven v's, eight w's, two x's, three y's, & one z.

More and More Pangrams

Looking back on it, I suppose the failure of the Mark I machine to find the pangram was a piece of good fortune. I mean, otherwise, the fast and flexible research instrument realized in the Mark II model may never have come into being. As it was, I could now experiment at will, initially confined only to the spectrum of possibilities defined by the given set of number-word ramges. This was an important limitation, since pangram-oriented ranges are unlikely to prove fertile in canvassing for autograms in general. In a self-enumerating *pangram*, the noncritical letters a, b, c, d, j, k, m, p, q, and z are likely to be prefixed by the words *one* or *two*; the frequency of o's, n's, e's, t's, and w's is thereby significantly slanted. Save in special cases, non-pangrams would give rise to distributions lying outside the scope of the machine.

The exploration I now embarked upon was a source of great fun and interest. A thoughtful Platonist can only wonder at some of the eternal Truths that God has seen fit to leave scattered about in the regions transversed by the machine. An early find was a somewhat wry specimen I couldn't resist sending off to Rudy Kousbroek. I suppose it might be decribed as a dead pan-gram:

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

Doubtless he will find little difficulty in producing a magic Dutch translation of this sentence. Another example which seemed worth drawing to his attention was:

This pangram containeth five a's, one b, two c's, two d's, twenty-five e's, seven f's, two g's, four h's, ten i's, one j, one k, one l, two m's, twenty n's, sixteen o's two p's, one q, five r's, twenty-six s's, twenty-one t's, three u's, six v's, ten w's, four x's, five y's, and one z.

I don't know wheteher he believed my tale of it having turned up among the marginalia in a folio edition of *Macbeth*. Probably not. The Dutch have never entirely succeeded in shaking off the legacy of German Scepticism.

If the above squibs suggest frivolity, it must be put down to the sudden release of tension after months of unrelenting effort. To have sought so long and so hard for a single jewel only to end up with a embarrassment of riches was an unhinging experience. For a while I reconnoitered without any clear plan. Among other diversions, sentences incorporating names of friends provided entertainment. It was interesting to find how readily some of these lent themselves to immortality:

This pangram for Doug Hofstadter contains five a's, one b, two c's, three d's, twenty-seven e's, seven f's, three g's, six h's, ten i's, one j, one k, one l, two m's, twenty n's, sixteen o's, two p's, one q, nine r's, thirty s's, twenty t's, four u's, six v's, seven w's, four x's, five y's, & one z.

In this way many pangram were unearthed, and the data derived from them shed new light on the relation between initial text values and the ranges in which solutions could be expected. This information could be plugged back into the machine through altering ranges so as to maximize the probability of future success with certain texts. After a time, the facility achieved in prospecting for nuggets prompted an ambitious new research program.

A shortcoming of logology, I find, is its absence of underlying structure. Like mathematics, it manifests itself in precisely defined chains of atomic symbols, yet lacks the intrinsic patterning, the symmetry of the former. Anyone with a feeling for mathematical form will probably regret this deficit too. Autograms, however, embody a peculiar fusion of both fields, an improbable marriage of arbitrary convention with arithmetical necessity. The unexpected possibilities they point to re-echo mathematical affinities. In particular, among other higher-order entities now appearing over the horizon of this strange realm are the counterparts of *numerical series*. The most obvious of these now became the focus of machine investigation:

This first pangram has five a's, one b, one c, two d's, twenty-nine e's, six f's, four g's, eight h's, twelve i's, one j, one k, three l's, two m's, nineteen n's, twelve o's, two p's, one q, eight r's, twenty-six s's, twenty t's, three u's, five w's, nine w's, three x's, four y's, and one z.

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

This third pangram contains five a's, one b, two c's, three d's, twenty-six e's, six f's, two g's, four h's, ten i's, one j, one k, two l's, two m's, twenty-two n's, seventeen o's, two p's, one q, seven r's, twenty-nine s's, twenty-one t's, four u's, six v's, eleven w's, four x's, five y's,a nd one z.

Prolongation of the series, written out in full, would be too space-consuming. Figure 6 presents an abbreviated record of the first twenty-five terms, with figures standing in for number-words. Note the use of a distinct verb in each case. This is not always necessitated, since the same word combined with different ordinals may also generate solutions. The employment of a different verb each time seemed to me demanded on aesthetic grounds.

The uncovering of this series is, in my opinion, among the most felicitous results of the machine. Though a mere matter of patient search, hundreds of running hours were involved. In one case, more than forty verbs were tried before locating a solution. On the average, though, winning combinations can be found for one in eight initial texts. This figure is empirically derived, of course. It seems to me worth pondering that, to my knowledge, no existent mathematical technique is able to assign even a rough value to the

probability of detecting a solution. Conceivably, artificially constructed number-word systems might be of use in gaining further insight into this.

The list here published is not as long as I could have made it. Eventually, I hope, one hundred will be reached. In the meantime, I can't help wondering how the discovery will strike others. Who could have forseen such a possibility? Once upon a time it had seemed daring to believe a single gem might exist. The finding of such a potentially infinite cluster of matching stones by far exceeds my greediest imaginings.

As I went along, I made up some new plug-in matrix cards using different resistor sets so as to cast a wider net, able to embrace certain kinds of non-pangram autograms:

This sentence employs two a's, two c's, two d's, twenty-eight e's, five f's, three g's, eight h's, eleven i's, three l's, two m's, thirteen n's, nine o's, two p's, five r's, twenty-five s's, twenty-three t's, six v's, ten w's, two x's, five y's, and one z.

The apparent elegance of these can sometimes be deceptive; closer scrutiny may reveal imperfections. For instance, oughtn't "one z" to be regarded as a redundant curlicue? Its inclusion is clearly a gratutitous addition to the preceeding text. Romantics may gaze indulgently at such ornament, but purists will point out that its real function is to contribute an extra o, n, and e merely in order to make the sentence work. Appending number-words is just a cunning way of disguising text-doctoring. Perhaps those with a sneaking affection for the solitary z will find consolation in:

This sentence contains three a's, three c's, two d's, twenty-six e's, five f's, three g's, eight h's, thirteen i's, two l's, sixteen n's, nine o's, six r's, twenty-seven s's, twenty-two t's, two u's, five v's, four x's, five y's, and only one z.

Here the inclusion of "only" legitimizes the addition of "one z" by "proving" it was premeditated. Even so, the choice of letter remains arbitrary: a q would have done just as well. Classicists, however, will reject all q's (whether straight or curly) and rightly insist on the crisp parsimony of:

This sentence employs two a's, two c's, two d's, twenty-six e's, four f's, two g's, seven h's, nine i's, three l's, two m's, thirteen n's, ten o's, two p's, six r's, twenty-eight s's, twenty-three t's, two u's, five v's, eleven w's, three x's, and five y's.

It is odd to reflect that the existence of this minimal form seems to vitiate the objection raised against the first version; "one z" may be redundant, but it couldn't have been thrown in just to make the sentence work! Subtleties of this kind should be kept in mind when trying to assess the relative merits of different specimens.

Bimagic Pairs and Banaagrams

At a still later stage, I constructed a second set of matrix cards representing number-words in Dutch. Besides another series of ordinal pangrams, one of the fruits of this excursion into a new language was:

Dit autogram bevat vijf a's, twee b's, drie d's, zevenenveertig e's, zes f's, vijf g's, twee h's, veertien i's vijf j's, een k, twee l's, twee m's, zeventien n's, twee o's, een p, een q, zes r's, vierentwintig s's, achttien t's, twee u's, elf v's, negen w's, een x, een y, en vijf z's.

Happily, this furnishes the first-ever truly impeccable magic translation, an earlier find being:

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

Notice that "en" is now reproduced as a fully-fledged "and". Strictly, it is inaccurate to speak of a *translation* in such cases, since the number-words themselves are not (in general) preserved. A preferable expression might be *transcription.* Another point, you might say, is that translations are inherently interpreter-dependent, whereas it is hardly likely that personal preference world influence an outcome here.

Representation of 25 Pangrams

This N^{th} pangram	-------	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1st	has	5	1	1	2	29	6	4	8	12	1	1	3	2	19	12	2	1	8	26	20	3	5	9	3	4	1
2nd	totals	5	1	2	3	29	6	4	7	10	1	1	2	2	21	16	2	1	8	28	23	4	4	9	3	5	1
3rd	contains	5	1	2	3	26	6	2	4	10	1	1	2	2	22	17	2	1	7	29	21	4	6	11	4	5	1
4th	numbers	4	2	1	2	29	7	2	6	10	1	1	1	3	23	14	2	1	9	26	20	5	5	9	3	5	1
5th	embraces	5	2	2	2	29	7	3	8	10	1	1	1	3	20	13	2	1	9	26	24	3	4	10	2	5	1
6th	harbours	5	2	1	2	28	7	4	7	10	1	1	1	2	21	15	2	1	7	28	20	4	6	9	3	5	1
7th	counts	4	1	2	2	30	5	3	7	9	1	1	1	2	23	16	2	1	7	28	21	4	7	9	2	5	1
8th	tallies	5	1	1	2	30	5	3	7	10	1	1	3	2	20	14	2	1	6	27	21	2	7	9	2	5	1
9th	exploits	4	1	1	2	28	7	4	8	13	1	1	2	2	22	16	3	1	9	26	23	5	4	9	4	5	1
10th	features	5	1	1	2	28	8	5	6	12	1	1	2	2	18	14	2	1	6	27	20	5	6	9	4	4	1
11th	utilizes	4	1	1	2	31	7	4	7	11	1	1	4	2	20	15	2	1	8	29	18	6	6	7	3	4	2
12th	tables	5	2	1	2	26	6	2	6	11	1	1	4	2	17	13	2	1	7	30	20	3	6	9	5	4	1
13th	includes	4	1	2	3	29	8	4	8	12	1	1	3	2	20	14	2	1	9	25	24	6	5	10	2	5	1
14th	recruits	4	1	2	2	28	8	4	7	10	1	1	1	2	20	15	2	1	10	26	24	6	3	9	3	5	1
15th	uses	4	1	1	2	30	7	2	5	9	1	1	1	2	22	16	2	1	5	27	21	3	7	10	2	5	1
16th	subsumes	4	2	1	2	30	7	4	8	10	1	1	1	3	21	15	2	1	8	29	21	6	4	7	3	5	1
17th	tabulates	6	2	1	2	28	7	3	5	10	1	1	2	2	20	14	2	1	6	29	24	5	6	10	4	5	1
18th	manifests	5	1	1	2	35	8	5	10	13	1	1	1	3	21	14	2	1	8	26	24	3	7	7	2	5	1
19th	assembles	5	2	1	2	35	6	5	10	12	1	1	4	3	18	12	2	1	8	28	23	3	7	9	2	4	1
20th	summons	4	1	1	2	29	7	3	5	11	1	1	2	4	22	16	2	1	6	28	21	5	6	10	4	5	1
21st	shows	4	1	1	2	29	6	3	6	11	1	1	3	2	22	16	2	1	8	29	21	4	4	11	5	6	1
22nd	displays	5	1	2	4	33	5	3	9	12	1	1	3	2	21	13	3	1	9	28	25	2	6	10	2	7	1
23rd	produces	4	1	2	4	26	6	2	4	10	1	1	2	2	22	17	3	1	9	29	21	6	4	11	5	6	1
24th	evinces	4	1	2	2	26	6	2	4	9	1	1	2	2	21	17	2	1	7	30	20	5	7	11	4	6	1
25th	discloses	4	1	2	3	32	7	3	9	11	1	1	3	2	20	14	2	1	9	28	25	3	5	10	2	6	1

Figure 6

In the actual pangrams, the numbers in the first column would be replaced by “first”, “second”, …, “twenty-fifth”. The numbers in the main body of the table would also be replaced by number-words. The fourth word of each pangram is shown in the second column.

Hardly likely, yet the local curvature of logological space can warp judgment much as it can warp a sense of humour. Here, for instance is a different English rendering of the same Dutch sentence, which is nevertheless another flawless magic transcription:

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

Sceptics may care to verify this assertion, barely credible at first sight. Once you have done so, it will be clear that even *magic* translations may depend upon the whim of an interpreter.

What is disturbing here is that the two English autograms, although differing in the number-words they use, exhibit indistinguishable texts. Or, to put it the other way around: although identically worded, the sentences list different numbers of letters. Certain minds seem to balk at this confrontation with a single text composed of thirty-one e’s this time and twenty-six the next. I have even known the delight of hearing someone patiently explain to me that such a thing can only be a patent logical impossibility.

Logic, however, should never be confused with logologic. The pair of autograms above is of course no more than a single text to which two solutions have been found. In concrete terms: having halted at a first solution, the machine was set running again so as to examine all remaining combinations and in this case succeeded in finding another one. The possibility of such *bimagic* sentences had been in my head from the first. Little did I dream that such a pair might also have a magic Dutch translation. As usual, though, the unexpected bonus is only a spur to greed, and one ends up regretting that the foreign version is not bimagic too. Discovery of a magic quadruple is an obvious goal for future research.

Though at first sight twisty, the cunning interlock between bimagic pairs is neatly brought out through a rather whimsical example:

This angram contains four a’s, two b’s, two c’s, one d, twenty-seven e’s, eight f’s, four g’s, five h’s, ten i’s, one j, one k, one l, two m’s, twenty n’s, fifteen o’s, one q, six r’s, twenty-seven s’s, eighteen t’s, five u’s, six v’s, seven w’s, three x’s, four y’s one z, but no —.

This angram contains four a’s, two b’s, two c’s, one d, twenty-seven e’s, eight f’s, four g’s, five h’s, eleven i’s, one j, one k, two l’s, two m’s, twenty n’s, fifteen o’s, one

q, six r's, twenty-seven s's, nineteen t's, five u's, six v's, eight w's, three x's, four y's, one z, but no —.

Abstracting the non-overlapping items for comparison shows:

ten i's	**eleven i's**
one l	**two l's**
eighteen t's	**nineteen t's**
seven w's	**eight w's**

The four numbers on the right are (by coincidence) all one greater than those on the left: a difference of one i, one l, one t, and one w. Cancelling common letters in the two listss will leave precisely that: the text on the right contains an extra i, l, t, and w. Differences at the meaning level exactly parallel those at the typographical level. Replacing one list with the other is thus an autogram-preserving change. A similar but more complicated pair of lists can be extracted from the previous example.

Notice that despite suggestive associations, a pair of sublists so derived can never comprise true *anagrams* (they cannot contain exactly the same letters). The letter content being identical, the *numbers* named could only be the same, and this is not the case. Taking into account both their slippery character and the *ban* on *anagrams*, I propose a special name for these curiosities: *bananagrams*. Beside their occurrence in bimagic autograms, a search for bananagrams could easily form a separate study in its own right.

How rare are bimagic cases? Of the roughly one in eight initial texts to yield a simple autogram, again something like one in eight of these turn out to have dual solutions. Is this coincidence, or might a theory be developed for predicting it? One might suppose the frequencies will change with different kinds of text, and yet experiments in Dutch give very similar results. Trimagic autograms and their associated trimagic bananagrams are naturally even rarer. Several hundred runs with the machine have located only one (with the unstimulating text, "This twenty-first pangram scored …") A finer example of the polymagic genre is:

This pangram tables but five a's, three b's, one c, two d's, twenty-eight e's, six f's, four g's, six h's, ten i's, one j, one k, three l's, two m's, seventeen n's, twelve o's, two p's, one q, seven r's, twenty-nine s's, twenty t's, five u's, six v's, eight w's, four x's, four y's, and one z.

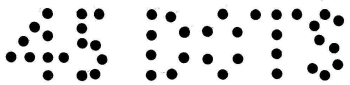
But this pangram tables five a's, three b's, one c, two d's, twenty-nine e's, six f's, six g's, eight h's, eleven i's, one j, one k, three l's, two m's, seventeen n's, fourteen o's, two p's, one q, eight r's, twenty-eight s's, twenty-two t's, six u's, four v's, eight w's, four x's, four y's, and one z.

The false modesty of the first is countered by the second one turning the tables!

So much then for the products of the pangram machine. Far from everything has found room for inclusion here. Aside from space considerations, the charm of such baubles is limited, one autogram soon seeming much like another. A few enthusiasts will continue to find fascination, I suppose, and indeed new topics in logology remain to be explored. One can only surmise what developments the future may reveal. Perhaps the magic sentences to come will possess a potency beside which these early essays in the craft will pale. That is certainly to be expected.

Among many possibilities that suggest themselves to logophiles is the extension beyond letter-level autograms to those enumerating every sign employed. There is a point worth raising in this connection. In the example shown at the start of this article, the listing of signs uses full names such as “comma” and “hyphen”. Seen retrospectively, this now seems less expedient than bringing them into line with the letters by reproducing the sign itself and adding an “ ‘s ”. Differences in British and American usage are among the recommendations for this change. Strictly speaking, however, quotation marks (or points) are demanded in using a sign as a name for itself. When this is done the apostrophe can be dispensed with and we arrive at: “… five ‘ a ’s, two ‘ b ’s … one ‘ z ’, twenty-seven ‘ , ’s, twenty-three ‘ ’ ’s, twenty-three ‘ ’ ’s, seven ‘ - ’s, &, last but not least, two ‘ & ’s,” for instance. This is, I believe, the most natural and formally correct method, and I recommend it as a notational standard to be adopted by others. The desirability of a universal system will apparent to interested parties.

Having said that, it is worth noting that the impulse toward sign-enumerating texts comes from a striving for completeness. This ambition can be fulfilled so long as conventional signs are treated as the atomic constituents of printed text. Atoms can be split, however, much like hairs. Reductionists will see the dot over the j as a typographic electron spinning in jeostationary orbit above its nucleus. As such, it will qualify for separate listing. Idealists will insist that ligatures were made in Heaven, and what God hath joined may no man tear asunder. Still others may contemplate descent to more hellish levels:



Perhaps my hesitation in giving an exact definition of the term “autogram” will now be more explicable. On consideration, it is probably a good idea to confine use of the expression to normal practice and leave the subatomicists to invent their own labels.

Aside from practical constraints, the initial text used in searching for an autogram is the sole determinant of success or failure. Time was when rambling and even dubious phrasing passed muster. Kousbroek’s pangram has changed all that; prolix or otherwise suspect formulations can no longer expect uncritical acclaim. At the other pole, however, is the prospect of zero-text autograms — simple, self-enumarating lists without even the “and” at the end. Since the ten non-critical letters are excluded, an inventory of this kind would comprise at most sixteen items. The shortest such list will in a sense be the ultimate autogram.

Also relevant in this context, though of less interest to logophiles perhaps, are *self-enumerating numbers*. A digit can never be catalogued as occurring zero times, so " 0 " can be used as a quotation mark to distinguish use from mention:

9000302020302090

— that is to say, nine zeroes, three twos, two threes, and two nines. On analogy with pangrams, pandigits can be found too:

2100070104020103020401050106020701080109 0

The 0-convention is admittedly arbitrary, but even if rationalized it would be hasty to suppose these oddities of any mathematical significance.

Still further contingencies for the future are metamagic autograms in which both words and letters come up for self-enumeration. More complicated monsters will present themselves to thought. Less fanciful are pairs of mutually-enumerating texts or even longer loops, although the difficulties these impose should not be underrated. A dyad such as

The sentence on the right contains… The sentence on the left contains…

cannot be handled independently. In effect, a magic combination must be found involving twice as many terms. Even so, the second sentence is a straightforward function of the first (or vice versa) , so that the problem need not imply construction of a machine having twice as many channels. I leave it to readers to explore the ramifications of this interesting puzzle. This brings me to a final word on the pangram machine.

Disconcertingly, more than one person who has seen the machine seems to have thought that at root it is really a computer. That is a misunderstanding. The term *computer* is now well established. There is nothing in the pangram machine corresponding to a central processing unit, an arithmetic-logic unit, a memory, or a program.

In fact, as I subsequently discovered, the machine is a closer cousin to a mechanical "number seive" invented by D.H. Lehmer in the 1920s. His device shares two things in common with mine. One is the basic odometer mechanism which sees to it that *combinations* of parameters are systematically called up for testing. The other is a *parallel* monitoring system that signals the odometer to halt only if every parameter meets a certain (not necessarily identical) condition. In Lehmer's apparatus the former is a motor-driven set of non-concentric parallel gears with holes drilled at special points in their periphery. The monitoring system is a light beam and photocell arrangement which brakes the motor when an alignment of holes is detected. The positions of these holes represent various finite-arithmetic solutions to an equation. A combination of such cases can yield a general solution. Note well the condition to be satisfied here (hole present at a certain location); in the pangram machine the criteria to be met (agreement with claimed numbers) are themselves a function of the parameters. Readers interested in further

details of Lehmer's sieve will find an excellent and entertaining account in Albert H. Beiler's *Recreations in the Theory of Numbers* (Dover Books).

A Challenge

The fact that two people working independently on quite different problems should have evolved closely similar mechanisms for their solution is remarkable. It suggests that the principle involved may have yet broader application. Indeed, I would like here to advance the view that the self-arresting odometer technique deserves a wider familiarity. There is a certain class of brute-force search for which it is a fundamental algorithmic structure. That is not to say I am advocating the constrcution of purpose-built machines (however enjoyable that might be). My idea is that an electronic *combination sequencer*, as I propose calling it, might easily comprise a standard hardware unit for integration into a (parallel) computer. This is not the place to elaborate on the idea. Suffice it to say that such a union could combine the speed of the former with the flexibility of the latter to produce a universal machine capable of accepting search problems from very different domains.

The increase in speed that both (a later version of) Lehmer's device and the pangram machine show over a conventional computer is directly attributable to their *parallel-processing*. Of course, non-conventional or "super" computers using parallel-processing also exist. This is worth mentioning, since in *Scientific American* A.K. Dewdney has given wide publicity to a remark of mine which seemed less reckless in its original context within a letter to Martin Gardner: "I bet ten guilders (five American dollars) nobody can come up with a self-enumerating solution to the sentence beginning, 'This computer-generated pangram contains …' *within the next ten years*." Parallel-processors, I should like to emphasize, are excluded from this wager.

Human perversity being what it is, not improbably some will not rest until I have been made to eat those words (it is incredible how seriously some people can take such artless taunts). I can only hope a respectable interval will be allowed to elapse before someone succeeds. In fairness, it must be said that much of the data contained herein could be put to use in greatly narrowing the area of a brute-force search. Frankly, I have often wondered how far one might go in returning to the computer armed with the insights and information gleaned via the machine. Besides this, from the present perspective, it is clear that a cooler analysis of the problem at the very beginning would have saved me a great deal of frustration later. Furthermore, subsequent discussion with various mathematicians and computer scientists make it clear that I am very far from having explored all software approaches; in particular, a modified version of the iterative algorithm originally tried is widely regarded as holding great promise. Leaving aside the wager, my warmest encouragement goes out to any who might like to pursue this question. There still remain a host of pangrams yet to be produced in all the languages remaining. Of keener interest, though, will be to learn of any new approaches pioneered.

Closing Thoughts

An act of magic consists in doing what others believe impossible. Together with, say, magic squares and the marvellous tessellations of Maurits Escher, autograms are among a class of objects which achieve their magical effect through creating an unbelievable *coincidence*. In the first, the coincidence is between row and column sums; in the second, between figure and ground shapes: in the third, it is between a message and its medium. These three are all examples of what Sigmund Freud (of all people) would have called *over-determined* structures — over-determined because they embody the simultaneous satisfaction of independent (sets of) criteria.

Of course there is already a discipline whose concern is with the creation of over-determined textual structures: a highly technical field in which the distillation of meaning and the coalescence of form with content have ever been focal concepts. Its name is *poetry*. Let none suppose that anything but poetry has been our purpose here.

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

References

Dewdney, A.K. "Computer Recreations" *Scientific American*, October 1984, pp 18-22.

Hofstadter, D.R. "Metamagical Themas" *Scientific American*, January 1982, pp 12-17.

Kousbroek, Rudy, "Welke Vraag Heeft Vierendertig Letters?" *NRC Handelsblad*, Cultureel Supplement 640, 11 February 1983, p.3.

Kousbroek Rudy, "Instructies Voor Het Demonteren Van Een Bom," *NRC Handelsblad*, Cultereel Supplement 644, 11 March 1983, p.9.

Kousbroek Rudy, "De Logologische Ruimte," Amsterdam: Meulenhoff, 1984, pp 135-53.