COMPUTER RECREATIONS

A potpourri of programmed prose and prosody

by A. K. Dewdney

"'Take care of the sense and the sounds will take care of themselves.'"
—The Duchess, in *Alice's Adventures in Wonderland,* by Lewis Carroll

As the renowned Oxonian master of nonsense observed, semantics takes precedence over syntax in creative writing. Good literature is shaped by the meaning contained in a writer's ideas. Computers are not yet capable of ideas and so cannot take care of a composition's sense. Yet, as a number of contemporary programs show, computers can certainly take care of the sounds. But is it art? That is for the reader to decide.

Consider the writings of MARK V. SHANEY, a computer program created by Bruce Ellis (now at the University of Sydney) that is based on an idea of Don P. Mitchell of the AT&T Bell Laboratories. MARK V. SHANEY is not exactly a self-starter. The program must first read and reflect on someone else's work. It then produces a rambling and somewhat confused commentary on the work. As an example, I quote MARK V. SHANEY's reflections on reading an elementary chemistry textbook.

"Wash the blackboard. Watch it dry. The water goes into the air. When water goes into the air it evaporates. Tie a damp cloth to one end of a solid or liquid. Look around. What are the solid things? What are the only ones that take part in the clouds themselves. As clouds move air tosses the tiny droplets of water. Sometimes the water which leaves the body in the leaves of green plants. It is quite porous, almost like a sponge. But the walls of the hollow tubes and cells are very strong. Chemical changes take place when something burns . . . ."

The commentary is hardly different when MARK V. SHANEY digests a book on elementary mathematics.

"Why do we count things in groups of five. When people learned how to count many things, they matched them against their fingers. First they counted out enough things to match the fingers of both hands. Then they put these things aside in one quart. A giant-size bottle that will hold four quarts is a three-digit number . . . ."

From here on, the program's output becomes increasingly confused, as readers may see for themselves in the illustration on this page.

Although sense is conspicuously absent from MARK V. SHANEY's writings, the sounds are certainly there. The overall impression is not unlike what remains in the brain of an inattentive student after a late-night study session. Indeed, after reading the output of MARK V. SHANEY, I find ordinary writing almost equally strange and incomprehensible!

How does MARK V. SHANEY produce its remarkable works? The answer is rather simple. The program's name, a weak pun on "Markov chain," provides a clue. In abstract terms, a Markov chain is a sequence of symbols generated according to a table of probabilities. In the version relevant to MARK V. SHANEY's operation, each row of the table corresponds to a pair of symbols. The entries in each row consist of individual symbols, each paired with an associated probability. A sequence of symbols is generated by an algorithm that begins with a "chain" of two symbols and thereafter cycles through four simple steps.

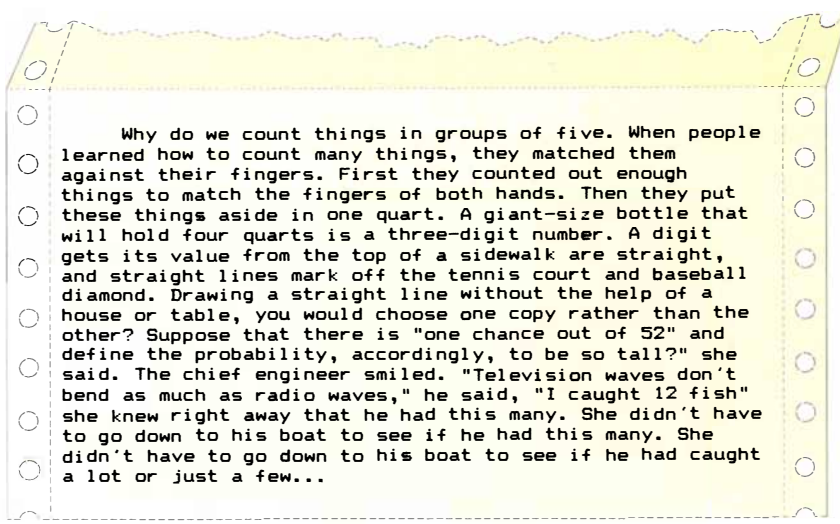
1. Find the last two symbols in the current chain.
2. Go to the row of the table corresponding to the symbol pair.
3. Select a symbol from the row according to its probability.
4. Add the selected symbol to the end of the chain.

For example, the first few entries of a Markov-chain table for the symbols *A, B, C* and *D* might look like this:

AB: *B*(.1) *C*(.1) *D*(.8)
AC: *A*(.1) *B*(.2) *C*(.1) *D*(.6)
AD: *B*(.4) *D*(.6)
BA: *B*(.3) *C*(.4) *D*(.3)
BB: *A*(.5) *C*(.5)

. . .

Given the symbol pair *AB* as the initial chain, the algorithm would have a 10 percent chance of selecting *B,* a 10 percent chance of selecting *C* and an 80 percent chance of selecting *D* as the next symbol. How does the algorithm choose a symbol according to the probabilities? It divides the interval between 0 and 1 into numerical



Why do we count things in groups of five. When people learned how to count many things, they matched them against their fingers. First they counted out enough things to match the fingers of both hands. Then they put these things aside in one quart. A giant-size bottle that will hold four quarts is a three-digit number. A digit gets its value from the top of a sidewalk are straight, and straight lines mark off the tennis court and baseball diamond. Drawing a straight line without the help of a house or table, you would choose one copy rather than the other? Suppose that there is "one chance out of 52" and define the probability, accordingly, to be so tall?" she said. The chief engineer smiled. "Television waves don't bend as much as radio waves," he said, "I caught 12 fish" she knew right away that he had this many. She didn't have to go down to his boat to see if he had this many. She didn't have to go down to his boat to see if he had caught a lot or just a few...

MARK V. SHANEY's mathematical commentary

segments whose lengths correspond to the symbol probabilities. It then chooses a random number between 0 and 1 and determines in which segment the number has fallen.

For row *AB* in the above table, the segments corresponding to the respective probabilities for *B, C* and *D* range between 0 and .1, between .1 and .2, and between .2 and 1. Suppose, then, that the computer's random-number generator yields .0172. Because that number lies in the first segment, *B* would be selected as the next symbol in the chain. The chain would then consist of the symbols *ABB*. The algorithm would next consult row *BB* in order to select a fourth symbol for the chain. Here again, a random number is generated. If it is less than or equal to .5, *A* is selected; otherwise the algorithm selects *C*. Because of its dependence on chance, if the algorithm were restarted with the same initial symbol pair, it might well produce an entirely different chain.

Such an algorithm was actually applied in the 1940's by Claude E. Shannon of Bell Laboratories to analyze the information content of human language. He constructed the algorithm's probability tables by scanning ordinary text and counting how many times each individual character followed each pair of characters (including blanks). Once the character frequencies for a given text were known, they could easily be changed into probabilities. The Markov chains of characters generated in this manner had statistical properties that resembled the source text, although they rarely contained valid words. How then does MARK V. SHANEY apply Markov chains to produce understandable English words?

The trick is to apply Shannon's algorithm for Markov chains but with entire words instead of characters as the concatenated symbols. As MARK V. SHANEY scans a text, it builds a frequency table for all words that follow all the word pairs in the text. The program then proceeds to babble probabilistically on the basis of the word frequencies.

A key feature of the program is that it regards any punctuation adjacent to a word as part of the word. That feature enables it to form sentences having a beginning and an end. Approximately half of them are even grammatical. I shudder to think what the program might produce after scanning this article!

Indeed, others have already shuddered at MARK V. SHANEY's reflections, some with rage and others with laughter. Some years ago Ellis decided to go on-line with his creation. The victims of the program's analyses were the innocent computer users who subscribed to an electronic bulletin board called net.singles. The bulletin board provides a place for male and female scientists, engineers, programmers and graduate students from all over the country to post their thoughts on subjects as diverse as dating, makeup and personal relationships. Why not have MARK V. SHANEY read the postings and respond with its own "thoughts" on those subjects? Here are some of MARK V. SHANEY's more thoughtful comments.

"When I meet someone on a professional basis, I want them to shave their arms. While at a conference a few weeks back, I spent an interesting evening with a grain of salt. I wouldnt dare take them seriously! This brings me back to the brash people who dare others to do so or not. I love a good flame argument, probably more than anyone . . . .

"I am going to introduce a new topic: does anyone have any suggestions? Anybody else have any comments experience on or about mixed race couples, married or otherwise, discrimination forwards or reverse, and eye shadow? This is probably the origin of makeup, though it is worth reading, let alone judge another person for reading it or not? Ye gods!"

The opinions of the new net.singles correspondent drew mixed reviews. Serious users of the bulletin board's services sensed satire. Outraged, they urged that someone "pull the plug" on MARK V. SHANEY's monstrous rantings. Others inquired almost admiringly whether the program was a secret artificial intelligence project that was being tested in a human conversational environment. A few may even have thought that MARK V. SHANEY was a real person, a tortured schizophrenic desperately seeking a like-minded companion.

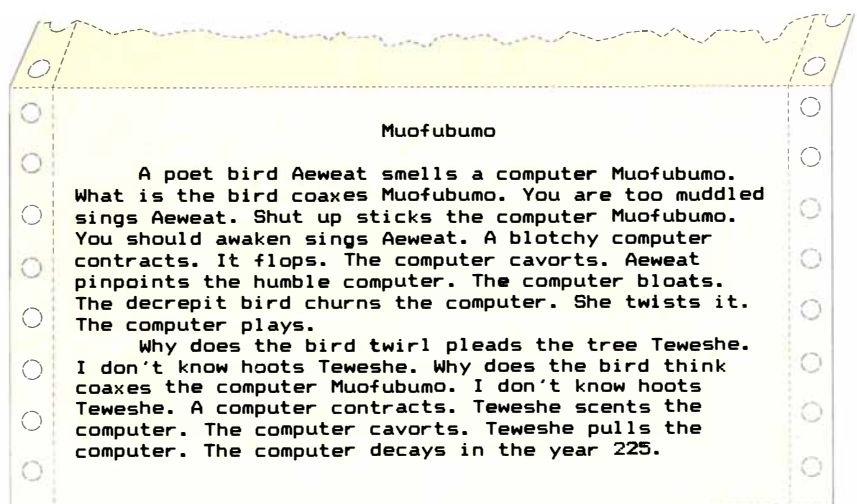
If the purpose of computer prose is to fool people into thinking that it was written by a sane person, MARK V. SHANEY probably falls short. MELL, the brainchild of Bonnie V. Firner of Piscataway, N.J., probably comes a little closer to that goal. MELL writes weird science-fantasy stories with a peculiar meditative quality.

"The warrior scowls in the drought pulses. He loves himself in the drought. He molds himself in the drought. He glowers at the warrior Dugaki in the drought pulses. He calls her in the drought. He snarls at her in the drought. He sits beside her beside awareness. He seizes her.

"Oh I am mighty says Oban. He smothers her in the drought. He smashes her of the turquoise . . . .

"Oban kills Dugaki. He has it of turquoise. He glares at it in the drought pulses. He calls it in the drought. He snarls at it in the drought. He sits with it beside greed. He seizes it. He smothers it in the drought. He smashes it of the turquoise. He needs it beneath the thunderbolt. His body burns beneath the thunderbolt . . . . "

MELL consists of some 1,500 lines of BASIC code; in contrast to MARK V. SHANEY, Firner's program is complicated even at the conceptual level. The program's main loop generates two characters in terms of 16 randomly generated "descriptors" whose values define qualities such as size, niceness, occupation, age and health, smell, commitment (from indifferent to fanatical) and even magical power.

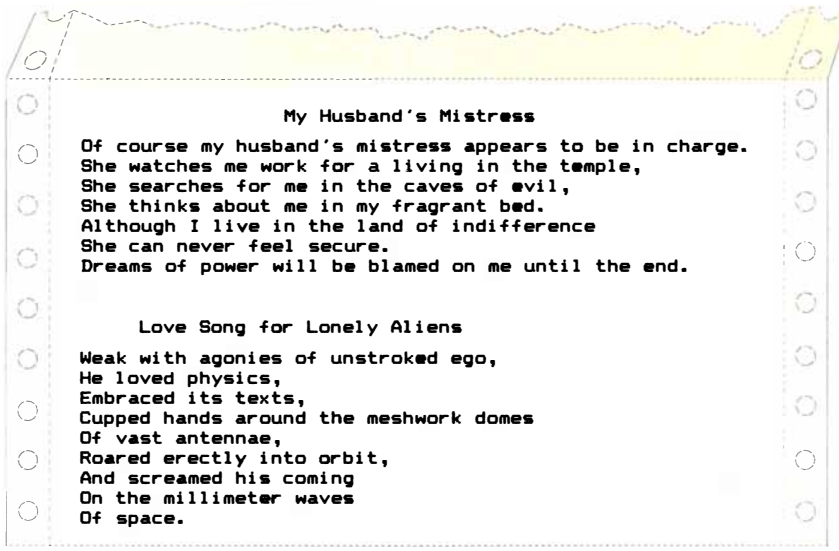


Muofubumo

A poet bird Aeweat smells a computer Muofubumo. What is the bird coaxes Muofubumo. You are too muddled sings Aeweat. Shut up sticks the computer Muofubumo. You should awaken sings Aeweat. A blotchy computer contracts. It flops. The computer cavorts. Aeweat pinpoints the humble computer. The computer bloats. The decrepit bird churns the computer. She twists it. The computer plays.

Why does the bird twirl pleads the tree Teweshe. I don't know hoots Teweshe. Why does the bird think coaxes the computer Muofubumo. I don't know hoots Teweshe. A computer contracts. Teweshe scents the computer. The computer cavorts. Teweshe pulls the computer. The computer decays in the year 225.

Firner's MELL tells a story



My Husband's Mistress

Of course my husband's mistress appears to be in charge.
She watches me work for a living in the temple,
She searches for me in the caves of evil,
She thinks about me in my fragrant bed.
Although I live in the land of indifference
She can never feel secure.
Dreams of power will be blamed on me until the end.

Love Song for Lonely Aliens

Weak with agonies of unstroked ego,
He loved physics,
Embraced its texts,
Cupped hands around the meshwork domes
Of vast antennae,
Roared erectly into orbit,
And screamed his coming
On the millimeter waves
Of space.

POETRY GENERATOR* (top) *and THUNDER THOUGHT* (bottom) *challenge the literati

Having chosen the names and traits of the story's characters, the main loop then determines what motivates a character by examining the values of its various descriptors. If one of the descriptors has a low value, MELL bases the character's interaction with the other characters on that fact. If no character has a low descriptor value, MELL will decide on the nature of a character's interaction based on his or her occupation. For each cycle of the main loop, the program then generates several sentences that describe the characters by fitting names for their qualities into predetermined grammatical slots. The sentences thus generated also include what Firner calls "background" words.

In the above story, for example, the background word "drought" sets the scene. Consequently that word creeps into many of the sentences. After generating a paragraph that describes an act by one of the characters in this way, the sentence-generating process starts all over again. Between iterations the program can change a quality of one of the characters. This helps to keep the story (as much as there is of one, anyway) from stagnating.

Is poetry any easier for a computer to generate than prose? One doubts it, because the meaning contained in prosody tends to be far more dense than in prose. Of the three versifying programs I shall discuss, only POETRY GENERATOR, which was written by Rosemary West of Mission Hills, Calif., is fully automated; the other two require human intervention to finish the product. West describes her program as follows:

"My approach... was to supply a large vocabulary of words and phrases that would be selected at random and combined according to a set of grammatical rules. For example, consider the following poem: 'The tree dips its bare fingers / into the black ice-pond / just as three gray geese / slide down a nearby snowbank.'

"Each line of the poem can be broken down into several parts.... 'The tree' is a noun phrase; 'dips' is a verb; 'its bare fingers' is the object of that verb. Having categorized the parts, I can then come up with between 100 and 400 possible substitutions for each part, one of which is randomly selected by the computer. For example, using the same verse structure, I might get: 'A woman hides five gray kittens / under the old jalopy / at the moment when the sad clowns / enter your museum of pain.' "

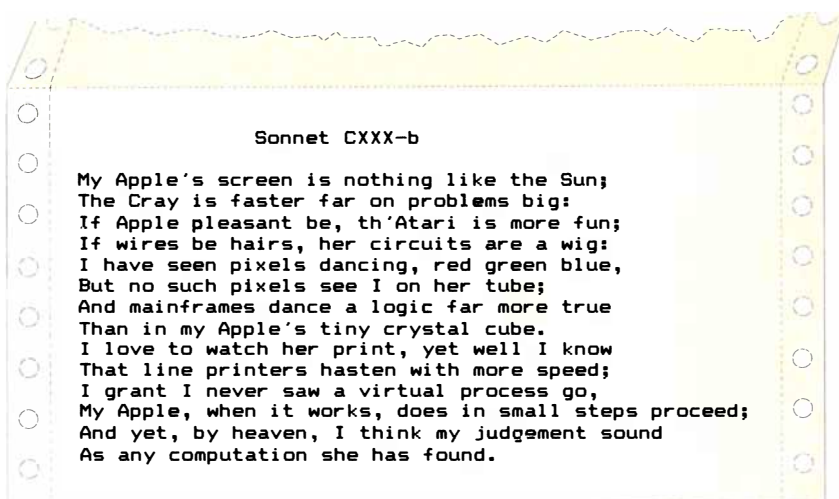
The poetic structures on which POETRY GENERATOR hangs its words may vary considerably, lending variety to the syntax and to what seems to be the meaning of the poem. West has based some of POETRY GENERATOR's output on the structures of her own poems, several of which have been published.

Thomas A. Easton of Belfast, Me., thinks the best way to generate computer poetry necessarily involves a symbiosis between human and machine. He has written a semiautomatic program called THUNDER THOUGHT that provides ideas for poems. Again, I quote the author of the program:

"I conceive of the creative mind as having two components: the popcorn mind and the critical mind. The former generates random combinations of whatever words, ideas and images happen to be in a sort of mental focus (along with peripheral material, which is why the popcorn mind can surprise us). The critical mind then discards as garbage the vast bulk of what the popcorn mind produces and edits, twists and elaborates the remainder to form poems."

Relying on internal lists of nouns, verbs, adjectives and adverbs, THUNDER THOUGHT operates roughly like West's POETRY GENERATOR, arranging words into sentence frames to produce what Easton regards as raw poetic material for a human mind to refine.

"The intermediate result is ungrammatical, nonsensical, ridiculous garbage... but not always. Among the many lines of garbage there always lie a few lines to which one responds. They make sense—or seem to. They beg one to tweak them a little. A pair of them insists that one make up a third line. They stimulate one to think of other lines that can accompany them.



Sonnet CXXX-b

My Apple's screen is nothing like the Sun;
The Cray is faster far on problems big:
If Apple pleasant be, th'Atari is more fun;
If wires be hairs, her circuits are a wig:
I have seen pixels dancing, red green blue,
But no such pixels see I on her tube;
And mainframes dance a logic far more true
Than in my Apple's tiny crystal cube.
I love to watch her print, yet well I know
That line printers hasten with more speed;
I grant I never saw a virtual process go,
My Apple, when it works, does in small steps proceed;
And yet, by heaven, I think my judgement sound
As any computation she has found.

An unknown author and ORPHEUS combine talents to compose a sonnet

A little editing, interpolation, elaboration and—voilà!—a poem."

Easton has written some 110 poems by this method and has published 32 of them—some have even appeared in literary journals. That ratio, he claims, is one that would turn many real poets green with envy.

The last word on computer poetry goes to ORPHEUS, a program designed by poet Michael Newman of New York City. ORPHEUS lays out strict frameworks, from haiku to sonnets, into which human writers may insert their own chosen words. Essentially ORPHEUS is a word processor ("poetry processor" in Newman's parlance) that lays out the lines of a given poetic form. The program allows a human being to fill in the lines according to whim and then to end them with the help of a rhyming dictionary. Setting ORPHEUS in sonnet mode, for example, one might write a pair of lines (in imitation of Shakespeare's 130th sonnet) as follows:

My Apple's screen is nothing like the sun;
The Cray is faster far on problems big:

Because the first line ends with the word "sun," ORPHEUS consults its rhyming dictionary and displays a number of words that rhyme with sun: bun, done, fun, gun and so forth. Scanning them, one's eye might alight on the word "fun." Is there a computer that is more fun than the Apple?

An association with games brings the Atari computer to mind for the next line.

If Apple pleasant be, th' Atari is more fun;

The first quatrain is polished off by selecting a word that rhymes with big.

If wires be hairs, her circuits are a wig:

The rest of the sonnet can be read in the illustration at the bottom of the opposite page.

The poetry programs I have mentioned may be bought by readers who want to sharpen their prosodic skills. Newman will be happy to supply ordering information for ORPHEUS, NERD (Newman's Electronic Rhyming Dictionary) and related products to those who write him at 12 West 68th Street, #2C, New York, N.Y. 10023. Easton meets West in a software package containing THUNDER THOUGHT and a program similar to it called VERSIFIER, which was written by West. Readers may inquire about the package by writing to Easton at Box 805, R.F.D. 2, Belfast, Me. 04915.

Readers may also be interested in what is perhaps MARK V. SHANEY's magnum opus: a 20-page commentary on the deconstructionist philosophy of Jean Baudrillard. That effort was directed by Rob Pike of the AT&T Bell Laboratories, with assistance from Henry S. Baird and Catherine Richards. The commentary can be obtained by writing Pike at the AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, N.J. 07974.

Students in Italy, stockbrokers in Singapore and physicians in the U.S. have joined the growing crowd of Mandelbrot-set devotees—all thanks to a ride on the Mandelbus, which I described in the February column. The effort to make the set's basic iteration algorithm understandable paid off in ridership and perhaps even readership. Yet, as a number of letters show, some confusion remains. It is only the Mandelbus' *first* stop that is being tested for membership in the Mandelbrot set. Subsequent stops may be either inside or outside the set, but if any of them turns out to be more than two units away from the origin, the first stop can be automatically excluded from the Mandelbrot set.

Readers who attempted to visit the area of the Mandelbrot set that Andrew LaMance calls Love Canal were disappointed to find empty space: they were given a wrong coordinate address. A minus sign placed in front of the first coordinate given should put readers squarely on the site of the curiously alluring pollution.

I may have been overly impressed by the magnification of 54,000 that Ken Philip of Fairbanks, Alaska, achieved in generating an image of a sea-horse scepter. Such magnification hardly marks the limit of a computer's acuity. Indeed, magnifications of that order are nearly routine both for A. G. Davis Philip of Schenectady, N.Y., and his brother Ken. The Schenectady Philip writes, "While I was in Fairbanks in November, my brother's Mac II produced a picture of a [Mandelbrot] midget at a magnification of 2×10^{31}. I consider that remarkable."

Peter Garrison of Los Angeles coincidentally explains that the double-precision mode available in most personal computers—in which the number of bits in the computer's "words" are doubled—actually more than doubles the computer's resolving power. In fact, the power is doubled for each one-bit increase in effective word size. Extremely high magnifications can therefore be achieved by means of the even greater precision made possible by special hardware and software.

Near the end of the column I mentioned a fast new algorithm for computing the Mandelbrot set that was described by Uval Fisher in the book *The Science of Fractal Images*. William S. Cleveland of the AT&T Bell Laboratories wrote to explain that the algorithm was actually developed by William P. Thurston and Allan R. Wilks. According to Cleveland, the new algorithm not only is faster but also produces more accurate pictures of the set than the standard algorithm does. As Cleveland says: "If you get on board the bus with the Thurston-Wilks algorithm painting the scene (in black and white), a wholly new and more realistic world will open up to you."

A new way to render beautiful Mandelbrot images using the traditional method was communicated to me by Carl G. Nugent of Seattle. It is now possible to see the set in delicate bas-relief, looking like the compressed fossil of an alien life-form. Although shaded in a single color, the images are just as beautiful as the full-color treatments because of their incredibly tactile nature: one can "feel" those delicate tendrils. The appearance is based on a trick that makes the tiny, intricate details of the set look from afar as though they were illuminated from one side.

The underlying idea is to divide the display screen into diagonal rows of pixels that run from the top left to the bottom right of the screen. If the iteration count generated for each pixel is taken to represent the pixel's "height," then an imaginary light source in the top left corner of the screen will cause a black "shadow" to be cast on certain pixels, depending on the height of the neighboring pixel (above and to the left) in its diagonal row. In even diagonal rows a pixel is not to be colored black unless its neighbor's height is strictly greater than that of the pixel; in odd diagonal rows, however, a pixel is colored black even if its neighbor has an equal height. Hence, in Mandelbrot "plateaus" (areas where all pixels have an equal iteration count), the diagonal pixel rows will alternate in color. Up close the displayed plateaus have a checkerboard appearance. "To make it look gray," says Nugent, "just throw away your microscope!"

FURTHER READING

COMPUTER RECREATIONS. Brian Hayes in *Scientific American*, Vol. 249, No. 5, pages 16–24; November, 1983.