Can a Computer Really Play Winning Chess?

David H. Ahl

Back in the '50s when Elvis was making like a hound dog, *Victory At Sea* was NBC's top prime-time hit, and Shockley demonstrated the first transistor at Bell Labs, some optimistic researchers speculated that computers really ought to be able to think. At that time, it was generally accepted that one good measure of "thinking" was playing a good game of chess. I'm sure that many chess experts and amateurs wouldn't disagree with that view today.

Chess is the intellectual game *par excellence*. There is no random chance involved, just sheer intellect in a situation so complex that neither player can hope to understand it completely, but sufficiently structured that each can hope to outthink the other. Even after 200 years of exhaustive play and thorough analysis, the field is still ripe for further exploration and development. Thus, if one could devise a successful chess program, one would seem to have penetrated to the core of human intellectual endeavor.

Three researchers in particular delved into this problem with great fervor. They were, of course, Allen Newell and J. C. Shaw at Rand Corporation, along with Herbert Simon at Carnegie Institute of Technology, now Carnegie-Mellon University.

During their efforts to produce a good chess-playing program, they discovered a number of things. First of all, it's not easy. Indeed, it may not be possible at all to produce a program that can play at the master level. Why not? First (and the reason that chess-playing programs are so good as a measure of thinking) is that the possible number of moves is in the neighborhood of infinity, give or take a bit. (Actually, Claude Shannon estimated that there are something like 10^{120} possibilities, which doesn't help us too much, since there are only 10^{16} microseconds in a century!). Consequently, you can't store all the possibilities in memory of any kind, nor could you possibly analyze them all. The approach, therefore, is to "teach" the computer how to play chess, evaluate possible moves, and formulate a playing strategy.

To teach a computer, it is probably helpful to first evaluate how a human plays chess. Most players have either learned from experience, or been taught by another player or book, that beginning game moves are best made following proven approaches. These openings minimize one's vulnerability and hopefully create a strong position for the attack to be launched in the middle part of the game. These "relatively" few openings, about 20 or so, follow reasonably predictable courses of action for perhaps the first seven to ten moves. Thus the opening of the game can largely be played from memory — human or computer.

Peter Jennings, president of Micro-Ware Limited, is the co-creator of Microchess.

The middle part of the game gets a bit hairy. Moves aren't nearly as predictable and most players adopt a strategy of evaluation of each move by considering its consequences after several more moves on both sides. This is, of course, what separates the men from the boys, so to speak, with the expert player able to accurately evaluate the consequences of a move 12 to 15 moves later while most novices are able to look ahead only 2 or 3 moves.

Different players put different importance on various pieces and positions. However, there seems to be general agreement that at least five factors should be considered in evaluating a move: mobility, value of attacked piece, vulnerability (and value) of attacking piece, King safety, and overall board configuration. Clearly, you don't just "tell" the computer these things and then say, "OK, now play."

If the middle game is hairy, the end game is downright mind-boggling when each side has maybe six or seven pieces left. Sides of the board have largely lost their meaning and configurations are possible that may *never* have occurred before. A strong attack can lead to an impossible defense in the span of just eight or nine moves. In short, heuristics and textbook approaches begin to break down and each player tends to develop his or her

Microchess on the KIM-1 proves a formidable opponent to the unwary at various computer conferences (here at PC'77 in Atlantic City).

own individual strategy. What does one do with a computer program at this point? Good question, with as many answers as there are programmers.

The ACM and some other groups have been running computer-chess tournaments for years. David Levy, an international master, regularly beats the winning computer at the end, although the playing is getting awfully good. Many of these programs are written on big (read, gigantic) machines of the CDC 6600 class, although some are on minis (I recall a Nova that played extremely well in the 1975 tournament in Minneapolis).

If the middle game is hairy, the end game is downright mind-boggling when each side has maybe six or seven pieces left.

Given this background and the difficulty of the problem, what can we expect from a micro? A year ago, I would have been tempted to say, "Not much." But Peter Jennings has proven that statement quite wrong.

Peter wrote a program called Microchess for the KIM 1. After all, what else can you do with a bare board, not-very-expandable microcomputer with only 1K of memory? Only 1K — you must be kidding! But I'm not kidding. Of course, it has a rather powerful 6502 chip at the heart of the system.

Peter follows the general outline above. The opening game consists of nine moves on both sides, in which the computer plays from a table if possible. This table occupies about 200-300 bytes. From there on, the program looks ahead approximately three moves on both sides and evaluates the possible outcomes from the standpoint of mobility, value of piece under attack and value of pieces open to attack. The program only uses castling in the opening and cannot capture pawns *en passant* — minor limitations in my opinion.

Peter has recently converted the program to run on 8080 systems; however, it requires 4K of memory. Actually, 2K is for I/O so the equivalent 1K 6502 program occupies 2K on an 8080 system. Interesting.

The accompanying manual is quite thorough and includes a description of the program, a sample run, and appendix with details of converting the I/O routines to your particular system. Microchess is available on cassette for the SOL or on paper tape for other systems. An interesting feature is the ability to reverse the board at any time which, if done continuously, lets the computer play against itself.

How good is Microchess? Well it beat me, but I'm a rank beginner, so that's no measure. Against Dark Horse, the program which won 6th place in the last ACM computer chess championship, Microchess was up one piece after 26 moves, but faltered and lost in the end game. (Microchess does not have a separate end-game strategy.) I had hoped to have the results of the game between Microchess and Fidelity Electronic's Chess Challenger, but that will have to wait until later.

Microchess costs $13 for the KIM-1, $18 for 8080 systems on paper tape or $20 on SOLOS cassette. Contact Peter R. Jennings, Micro-Ware Ltd., 27 Firstbrooke Road, Toronto, Ontario, M4E 2L2, Canada. (416) 424-1413. ■

Another View of Microchess

A most interesting thing about Microchess is the manner in which it generates moves. Unlike most larger chess-playing programs, Microchess selects a move as a result of a sequential search through all possible moves. This very primitive algorithm for move evaluation does not hamper Microchess as much as one might guess. An example of typical play by Microchess may be seen in the following opening game:

Microchess (white)	*Challenger* (black)
KP - K4	KP - K4
KN - KB3	KN - KB3
KN x KP	KN x KP
Q - KN4	KN - QB4
QP - QP4	Q - KB3
QP x KN	Q x KN
K - Q2	Q x KP
Q - K4	Q - K2
Q - Q4	

Obviously neither player was playing particularly inspired chess; however, all the moves made by Microchess were reasonable if not optimal. Unfortunately Microchess does not fare so well when the game requires a move that is not obvious in the current board position as can be seen by the following opening:

Challenger (white)	*Microchess* (black)
KP - K4	KP - K4
Q - KB3	Q - KR5
KB - QB4	QP - Q4
KB x QP	QBP - QB3
Q x KBP	K - Q1
Q x KB	K - QB2
Q x KNP	KN - K2
Q x KP	K - Q1
Q x KR	

Another flaw in this program is the fact that the internal board representation used by Microchess will allow only one Queen per side at any time. Thus the value of pawn promotion is considerably reduced.

Richard Freeman
Irvine, CA

Macrochess is a popular way to spend an afternoon in Frankfurt-am-Main. While games are played between individual players, each has a team of advisors (hecklers?) ready with opinions and recommendations.