COMPUTER RECREATIONS

The king (a chess program) is dead, long live the king (a chess machine)

by A. K. Dewdney

If CRAY BLITZ had a memory for anything except the moves of chess, it would never forget the evening of October 15, 1985. It is the last round of the North American Computer Chess Championship, held at the annual meeting of the Association for Computing Machinery. A space at the front of a meeting room in the Radisson Hotel Denver is taken up by five tables, which are separated from the audience by a barrier. At each table two teams of programmers and advisers face each other. Sometimes they joke and sometimes their faces fall into the blankness of wondering and waiting. Behind each table is a display screen on which an overhead projector casts the image of a current board.

The tournament features 10 contenders for the North American title. Their names are odd and angular, betraying differing origins and aspirations: AWIT, BEBE, CHAOS, CRAY BLITZ, HITECH, INTELLIGENT SOFTWARE, LACHEX, OSTRICH, PHOENIX and SPOC [*see illustration on next page*]. Missing are three of the big names that have dominated computer chess in recent years: BELLE, CHESS 4.7 and NUCHESS.

Most of the interest is focused on the championship game between CRAY BLITZ and HITECH. On the CRAY BLITZ side of the table are Robert Hyatt of the University of Southern Mississippi, Albert Gower, a chess expert from the same institution, and Harry Nelson of the Lawrence Livermore National Laboratory. Facing them are Hans Berliner of Carnegie-Mellon University and Murray Campbell, one of his graduate students, who is an expert player. Berliner fills the dual role of chess adviser and programmer for the HITECH team. As the game wears on and tension mounts, Berliner rises often from the table, a weary smile on his face. Once he strolls past my chair and mumbles, "This is too much like my U.S. championship days!" (For several years in the late 1950's and early 1960's Berliner was rated among the top dozen players in the U.S.)

Unlike the U.S. championship, in which a deathly silence reigns, this tournament is filled with conversation, occasional laughter, the rattle of keyboards and a continuing microphone commentary by adjudicator Michael Valvo, a flamboyant computer consultant and international chess master from Sedona, Ariz. "A weak move by Black. The king is still too exposed and the doubled pawns on c5 and c6 continue to hamper the defense." Nearby a member of the CRAY BLITZ team exclaims to no one in particular, "That's funny, I thought it would play king to f3." An international master can still spot flaws in computer chess and programs still surprise their creators.

Throughout this final round of play it has been obvious that HITECH has the advantage over its rival: early in the game CRAY BLITZ has fallen into a zugzwang, a critical position from which any conceivable escape involves either a bad move or a loss of material. In this case CRAY BLITZ has been forced to structure its pawns badly. HITECH continues to exploit the advantage.

By midnight it is all but over. Most of the games are finished and the experts claim a win for HITECH. The CRAY BLITZ team asks adjudicator Valvo for permission to resign. He suggests two more moves: if the CRAY BLITZ position is no better then, the team may resign. It is not and they do. HITECH is North American champion and de facto king of computer chess. Although CRAY BLITZ is the official world champion (it won the title in 1983 and does not have to defend until June), HITECH's win, along with its three other tournament victories, is impressive. HITECH is almost certainly the world's strongest chess-playing computer.

There are smiles and more conversation. Did the absence of BELLE, CHESS 4.7 and NUCHESS make a difference? "It would have been nice if BELLE and some of the other programs could have made it," said one organizer of the tournament, "but I don't think the outcome would have been much different." He went on to point out that in terms of the programs and machines entered, there was no effective difference between the North American and the world championships. The talk turns to Kasparov and Karpov and then to theory. "I'm not kidding," says an apparently knowledgeable participant. "A 20-ply program that looks only at material can beat any grand master." There is some argument, but in a few more minutes the room is empty. The North American championship is over.

The claim about the 20-ply program is an interesting one. The game of chess can be represented by a vast tree consisting of nodes and lines. I visualize the tree upside down, so that the root node is at the top. Each node represents a possible position, namely a chessboard on which the pieces and pawns have arrived at their squares through legal play. A node is joined to a descendant node by a line if the move of a single piece or pawn converts the board represented by the former node into the board represented by the latter node. A game of chess can always be identified with a particular path through the chess tree from the root node (in which no moves have been made) down through the tree to some node where, as a general rule, few pieces are left and one player has been checkmated or forced to resign.

A chess-playing program attempts to explore only as much of the game tree as is necessary. From the node representing a current position it examines all the descendant boards (ply 1), examines the descendants of the descendants (ply 2) and so on. The average depth of its exploration is called the lookahead. This measure comprises the greater part of what might be called a chess program's intelligence. The lesser part arises in the program's evaluation of the boards constituting the horizon of its lookahead. It analyzes these boards and attaches a numerical value to each one. The value reflects the desirability of reaching that position. Using a procedure called minimax, the program causes some of the values thus assigned to percolate up the tree to the nodes at ply 1. The node receiving the highest value is the play to make.

There is an interesting tradeoff be-

Scientific American, February, 1986

tween the two parts of the program's intellection: the better its evaluation scheme is, the less deeply it needs to search the game tree. Indeed, if it had a perfect evaluation scheme, it would never have to search deeper than one ply. Conversely, a program with a very simple evaluation scheme must search much deeper if it is to play effectively. How deep must a search of the kind that looks only at material be in order to be effective against a grand master? Would a 20-ply search suffice?

The title of grand master is awarded by the Fédération Internationale des Échecs to players who distinguish themselves in international play. (The federation bars computers from consideration.) Grand masters generally have ratings higher than 2,400, the level of a senior master. Up to the time of the North American Computer Chess Championship, HITECH had played 21 games in human tournaments, earning a rating of 2,233. This made it the highest-rated chess-playing computer in the world. According to Berliner, who was rated at 2,443 in his competition days, HITECH's rating has increased by an average of eight points a game in national tournaments. Dare one suppose that in just 14 more games the machine will surpass its designer?

All of this raises the question of just how good chess-playing computers will eventually become. Will a computer ever be the best chess player in the world? David Levy, former player and present author-entrepreneur, has committed the question to a series of wagers. In 1968 Levy bet John McCarthy of Stanford University £500 that no computer would succeed in beating him in a chess match for the next 10 years. Levy collected in August, 1978, at the Canadian National Exhibition in Toronto. There he toyed with CHESS 4.5, a program created at Northwestern University. The basic bet was thereafter renewed in the amount of $6,000 for a period of six more years. In April, 1984, Levy in London played a telephone match with CRAY BLITZ. He won again.

Levy's short streak emboldened him to offer the following £100,000 wager in Denver: within 10 years of the offer each computer challenger will have been defeated by a human player selected by Levy. If Levy finds a taker, it will probably be not a mere program but a specialized computer. So far there have been no takers.

The two top finishers in the North American tournament, HITECH and BEBE, were essentially such chess machines. Interestingly, Levy's own entry, a program named INTELLIGENT SOFTWARE, came in third. It runs on an Apple IIe computer that features nothing more sophisticated than an accelerator board, which is a special circuit card that doubles the speed of the machine. Perhaps Levy has developed a superior evaluation scheme.

PROGRAM	ORIGIN	COMPUTER	LANGUAGE	BOARDS/ SECOND	LOOK-AHEAD
AWIT	University of Alberta	Amdahl 5860	Algol W	10	3-ply
BEBE (second)	SYS-10, Inc., Hoffman Estates, Ill.	Custom machine	Assembler	20,000	7-ply
CHAOS	University of Michigan	Amdahl 5860	FORTRAN	70	4-ply
CRAY BLITZ	University of Southern Mississippi	Cray X-MP 48	FORTRAN/ Assembler	100,000	8-ply
HITECH (first)	Carnegie-Mellon University	Sun with custom VLSI	C	175,000	8-ply
INTELLIGENT SOFTWARE (third)	Intelligent Software, Inc., London	Apple IIe with accelerator	Assembler	500	7-ply
LACHEX	Los Alamos National Laboratory	Cray X-MP 48	FORTRAN/ Assembler	50,000	7-ply
OSTRICH	McGill University	Network of seven Novas and an Eclipse	Assembler	1,200	6-ply
PHOENIX	University of Alberta	Network of VAX 780's and 10 Suns	C	540	6-ply
SPOC	SDI Cypress Software, San Jose, Calif.	IBM PC	Assembler	300	5-ply

Entrants in the 1985 North American Computer Chess Championship

The chess cognoscenti at the championship agree that the best game of the tournament was played during round two between CRAY BLITZ and BEBE, a product of private enterprise. Tony Scherzer, whose company SYS-10, Inc., developed BEBE, has transported his charge to a number of tournaments in recent years. BEBE is no mere program but a chess-playing machine in its own right. The game was significant not only because it was the most interesting of the tournament but also because it was the first time CRAY BLITZ had lost in three years.

Readers with a chessboard can follow the CRAY BLITZ V. BEBE game by playing the 50 moves listed below. Pieces are denoted by capital letters: K, king; Q, queen; B, bishop; N, knight, and R, rook. Chessboard squares are referred to by letter-number coordinates: when the board is in the standard position, so that the lower left-hand square is black, the files, or columns, are labeled from left to right with *a* through *h;* the ranks, or rows, are numbered 1 through 8, beginning at the bottom of the board. Notation employed in listed games such as the one below varies from the straightforward Kb1 (king to square b1) to the puzzling Nf3 (knight to f3). Which knight? On that particular move only one knight can jump to f3. A move by a pawn is indicated by the designation of a square, for example e4. The game is annotated by Valvo.

	CRAY BLITZ (White)	BEBE (Black)
1.	e4	c5
2.	Nf3	d6
3.	d4	cxd4

(The x means a piece or pawn is taken.)

4.	Nxd4	Nf6
5.	Nc3	g6
6.	Bg5	Bg7
7.	Qd2	Nc6
8.	0-0-0	0-0

(White castles on the queen side and Black castles on the king side.)

9.	Nb3	Re8
10.	Bc4	Ng4

Black played Ng4 intending 11... Bc3xN in the next move. Black may have thought that 12 c3xB is forced on White, but Black changes its mind on White's next play. If Bxc3, then Qxc3!; Nxf2 fails if either rook is played to f1, whereas Bxf7, a check, would be fatal.

11.	h3	Nge5
12.	Bb5	a6
13.	Be2	a5
14.	Bb5	Be6
15.	Nd5	a4
16.	Nd4	Bd7

White's situation is desperate. The Black pawn (a4) threatens to create weaknesses around White's king.

17.	Nxc6	bxc6
18.	Nxe7 check	Rxe7
19.	Bxe7	Qxe7
20.	Be2	Qe6
21.	Kb1	Rb8

(The chess board at this point in the game is shown below.) Black threatens 22...Rb2, a check, which is followed by 23 Kb2 Nc4, a fork that wins the White queen.

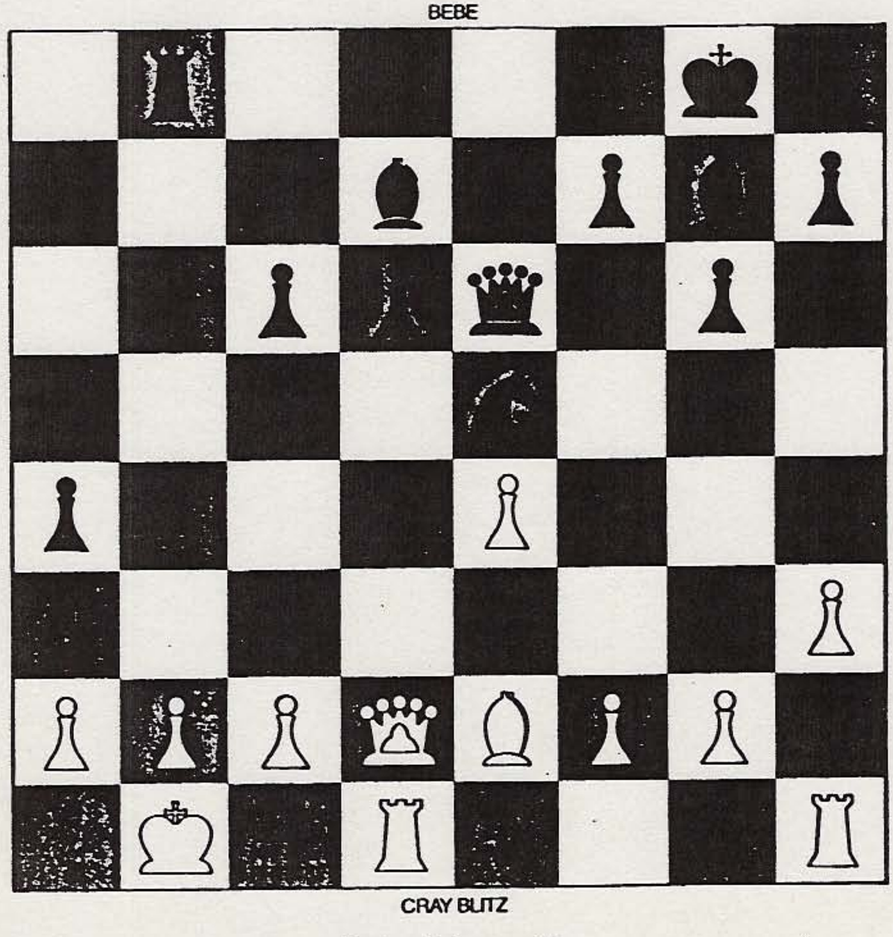
22.	b3	axb3
23.	cxb3	Be8
24.	Kc2	Nd7
25.	f3	Ra8
26.	Kc1	Nc5
27.	Qc2	Qf6
28.	Bc4	Qa1 check
29.	Kd2	Qxa2

An even stronger move is 29...Bc3, a check, followed by 30 Ke2 Ra2!

30.	Qxa2	Rxa2 check
31.	Kc1	d5
32.	exd5	cxd5
33.	Bxd5	Bb5
34.	Rhe1	Nd3 check

Black's material advantage of one piece is about to be increased by another exchange. In human tournament play White could reasonably resign at this point.

35.	Rxd3	Bxd3
36.	Re8 check	Bf8
37.	g4	Kg7
38.	Re3	Ba3 check
39.	Kd1	Ra1 check
40.	Kd2	Bf1
41.	Kc3	Rc1 check
42.	Kd2	Rc5
43.	Ke1	Bxh3
44.	Bc4	h5



The board after move 21

45.	gxh5	gxh5
46.	Kf2	h4
47.	Rd3	Bf5
48.	Rd4	h3
49.	Rh4	Rc7
50.	Rh5 (asks to resign)	

The CRAY BLITZ program runs on a Cray XM-P 48 computer. Famed for its speed as a multiprocessor, the Cray is nonetheless a general-purpose computer and not a chess machine. BEBE, whose circuits are devoted to chess playing, obviously outperformed the Cray–CRAY BLITZ combination in the game above.

HITECH is in a sense even more specialized. When Carnegie-Mellon University was the Carnegie Institute of Technology, a chess-playing program called TECH was developed there. The name HITECH reflects the fact that Berliner, Campbell and the other members of the HITECH team, Carl Ebeling, Gordon Goetsch, Andy Palay and Larry Slomer, have revived the TECH tradition in a world of very-large-scale integration (VLSI) and burgeoning parallelism. The HITECH machine combines a Sun computer equipped with a specially designed processor that Berliner calls the searcher. The Sun computer runs three programs: a user interface, a task controller and an oracle. The oracle embodies what computer chess experts call the book. This is a large catalogue of chess openings and variations that human chess experts commonly know. The oracle's data base contains a great deal of other chess knowledge that can be easily expanded. When the searcher examines the possibilities of play from a given position, it proceeds on the basis of chess information relevant to that position downloaded from the oracle.

The searcher itself contains a microprocessor and several hardware modules that generate moves, evaluate moves, check for repeated moves and so on. The microprocessor coordinates their activities. The move generator consists of 64 VLSI chips, one for each square of the chessboard. Each chip examines the entire board in order to determine whether any piece or pawn can be moved to the square under its purview. It determines the best move in terms of standard criteria such as opportunities for capture or control of the center. At the same moment the other 63 chips are doing the same thing. If there are 10 pieces on the board, this architecture means that possible moves are generated 10 times faster, other factors being equal.

The evaluation of moves must keep

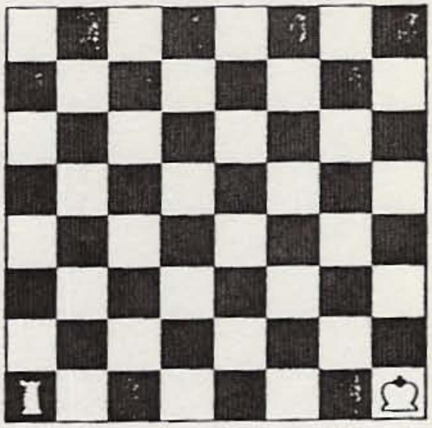
up with the generation process. A first stage of evaluation is carried out by the move generator itself. It houses a kind of supervisor that judges among the moves generated by the 64 chips. Each chip computes a number that estimates the strength of its best move and transmits the number to the supervisor. The chip-generated numbers are like cries for attention. The supervisor ranks them in order of loudness (read effectiveness).

HITECH then proceeds to search the game tree, following the ranking produced by the supervisor of the moves that are possible from the current position. A second stage of evaluation is performed by the evaluation module at each new position generated within the game tree. Using chess knowledge (downloaded from the oracle) relevant to the current position, the module evaluates each board, whether or not it is at the lookahead horizon. Such is the nature of parallelism. The additional effort costs no additional time. The Sun's task controller tells the searcher how deeply to search the tree and, when the search is completed, whether to probe still further. In this way HITECH manages an average lookahead of eight ply but will, on occasion, search as deep as 14 ply. This may seem a long way from the 20 ply that may be needed to beat a grand master. On the other hand, HITECH's use of parallelism and sophisticated deployment of chess knowledge into the tree search may compensate for the relative shallowness of the search. In any event, by the time of the next World Computer Chess Championship in June in Cologne, West Germany, HITECH may be unstoppable.

HITECH is the world's newest chess machine. The first such machine was invented in 1890 by Leonardo Torres y Quevedo, a Spanish engineer. Using mechanical levers, pulleys and electromechanical switches, it played a mean game of rook-and-king v. king. Human players were given the privilege of managing the affairs of a lone king seeking to evade checkmate by the machine's powerful combination of a rook and a king. Torres y Quevedo's machine always won.

Readers are invited to create a strategy that produces this result. It should be assumed that the human player's king does not begin life in a stalemate (unable to move without placing itself in check). Then the task is to specify in as few rules as possible how the machine achieves checkmate from an arbitrary position. The position shown in the illustration on this page is a reasonable starting point.

The machine plays White and is to move first. How do White's king and rook combine to drive Black's king into checkmate? White could begin by moving its rook to the *d* file. This would prevent Black from moving its king any farther to the left. The maneuver can be repeated if the Black king obligingly moves to the right, but what if it continues to occupy the *e* file, merely cruising up and down? I shall publish the most succinct solution, whether it is in algorithmic or in natural language.



How to checkmate with rook and king?

Last November's "Computer Recreations" featured flibs: finite living blobs that attempt to predict changes in their environment. In the primordial computer soup, during each generation the best predictor crosses chromosomes with a randomly selected flib. Increasingly accurate predictors evolve until a perfect one emerges.

A flib is essentially a finite automaton. That is, it has a finite number of states, and for each signal it receives (a 0 or a 1) it sends a signal and enters a new state. The signal sent by a flib during each cycle of operation is its prediction of the next signal to be received from the environment.

Some readers gave their flibs impossible prediction tasks. No flib will ever evolve that can predict a sequence of random bits. Nor will flibs ever develop to predict primes. It is perfectly reasonable to ask a flib to predict a repeating binary sequence. For example, there is a 4-state flib that will predict the repeating eight-symbol sequence 01100010. Even a repeating sequence, however, can tax the predictive abilities of a flib if its basic string is too long in relation to the number of states in the flib. As it happens, no 4-state flib will ever predict the repeating sequence 010010111. Why not?

The simplest answer to the question involves a process I call creeping induction. Imagine a 1-state flib. It might predict the endless repetition of the basic string 01. For each of the two possible signals the flib receives there is one response: if a 0 is received, the flib sends a 1 and then reenters the same state. If it receives a 1, it sends a 0. A basic string of three symbols, say 011, is beyond the ability of a 1-state flib to predict because the automaton simply does not have an adequate stock of responses. A 2-state flib, on the other hand, has four possible responses, two for each state. Thus it can predict a repeating string of four symbols but not one of five symbols; when the fifth symbol is reached, the flib must repeat an earlier response. The argument is clear. An n-state flib can predict a basic string that is $2n$ symbols long but not a string $2n + 1$ symbols long. There is some pleasant distraction to be had in devising a basic string eight symbols long and then constructing by hand the 4-state flib that will predict it. The perfect predictor thereby obtained is essentially unique. It is possible to measure the success of one's AUTOSOUP program by comparing the perfect predictor that evolves from it with the flib already constructed.

Several readers found ways to make AUTOSOUP run faster. For example, there is not much point in testing the current batch of flibs on a sequence of 100 environmental symbols if the basic string is only six symbols long. One repetition of the string will produce 12 environmental symbols, which should be enough for most purposes.

Philip Kaaret of Princeton University has pointed out that the program can also be shortened if two flibs rather than the entire population are scored on each execution of the main loop. After all, only two flibs (at most) have changed: the lowest-scoring flib has been replaced by a new hybrid, and one other flib has perhaps been struck by a cosmic ray.

The speedups obtained by shortening the environmental test sequence and by eliminating the test altogether for old flibs are roughly equivalent. Now there will be time to evolve n-state flibs that can predict repeated basic strings as many as $2n$ symbols long.

From his letter it appears that Ed Coudal of Park Ridge, Ill., was loath to send his lowest-scoring flib directly to the choir celestial. Instead he bred it with the highest-scoring flib at each cycle. By following this scheme Coudal could in fewer than 40 generations derive flibs capable of predicting a six-symbol basic string.