# COMPUTER V. CHESS-PLAYER

by ALEX BERNSTEIN and MICHAEL de V. ROBERTS

REPRINTED FROM
SCIENTIFIC
AMERICAN
JUNE 1958

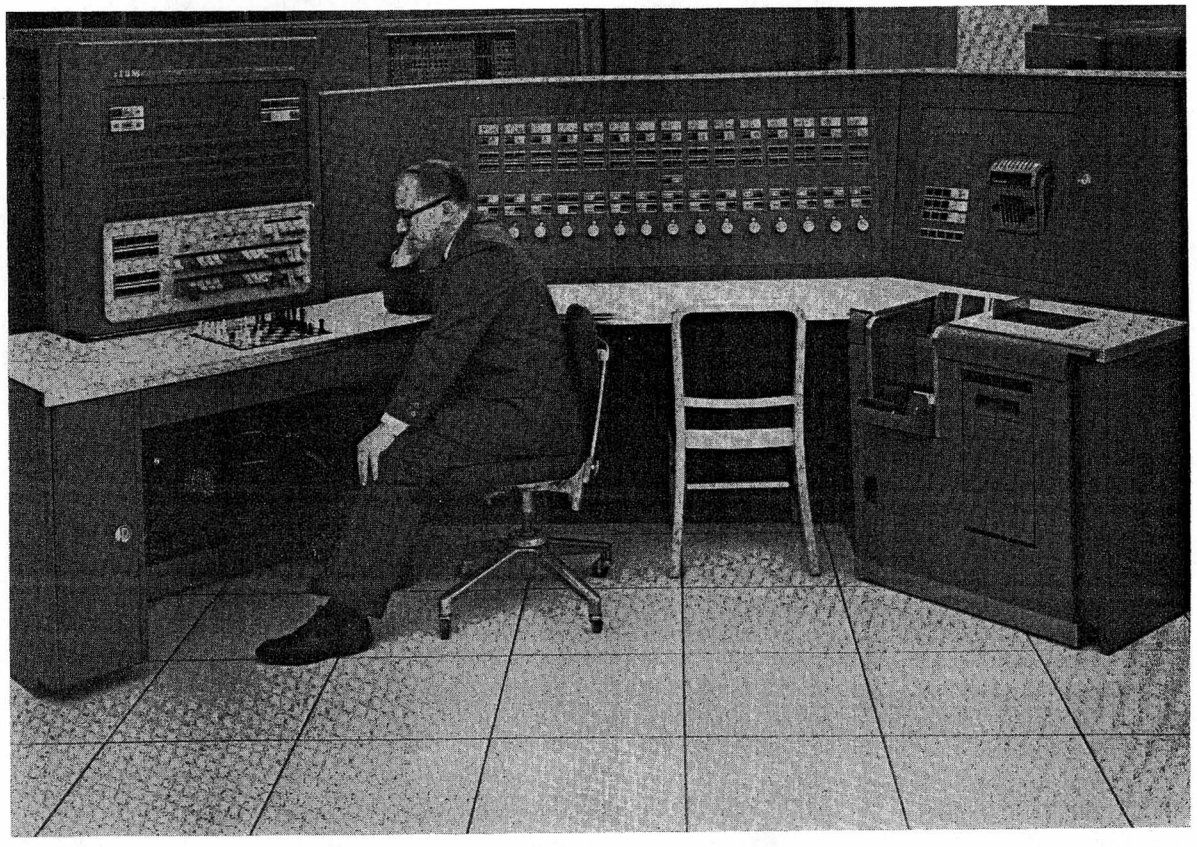
# Computer v. Chess-Player

*Can a machine be made to think creatively? One reply is: First, can a machine be made to play a good game of chess? How a computer was programmed so that it could defeat an inexperienced human opponent*

by Alex Bernstein and Michael de V. Roberts

Chess is not only one of the most engaging but also one of the most sophisticated of human activities. The game is so old that we cannot say when or where it was invented; millions of games have been played and thousands of books have been written about it; yet the play is still fresh and forever new. Simple arithmetic tells why. On the average, each move in chess offers a choice of about 30 possibilities, and the average length of a full game is about 40 moves. By this reckoning there are at least $10^{120}$ possible games. To get some idea of what that number means, let us suppose that we had a superfast computing machine which could play a million games a second (a ridiculous supposition). It would take the machine about $10^{108}$ years to play all the possible games!

So no conceivable machine could play a perfect game of chess, examining all possible moves. This is what makes the problem of programming a computer to play chess so intriguing. A present-day computing machine, with all its speed of calculation, is about as limited as a human being, on any reasonable time scale, in exploring the likely consequences of a chess move. Since it cannot study all the possibilities, the machine must play the game in human

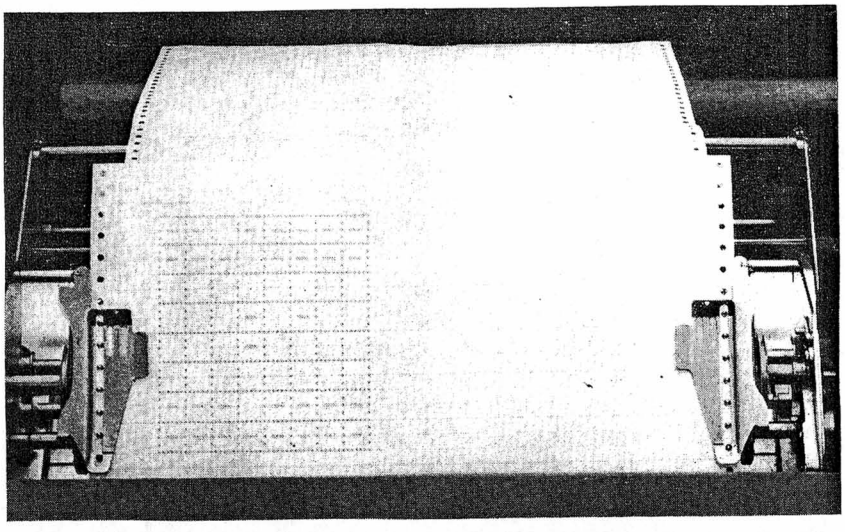


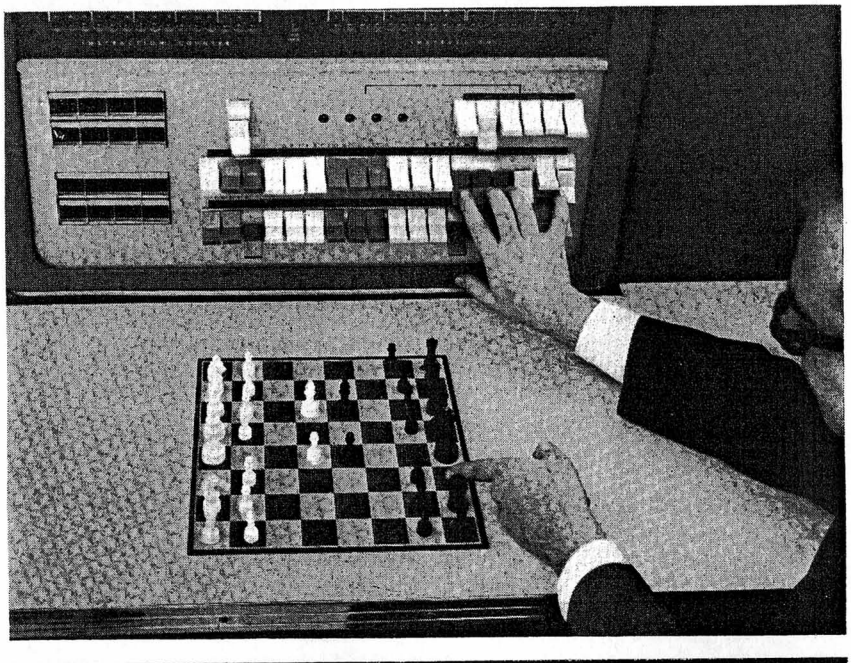
OPPONENTS IN CHESS GAME depicted here are Alex Bernstein, co-author of this article, and an IBM 704 computer. The game is played on an ordinary chessboard, but information about each move is fed into the machine by controls above the board.

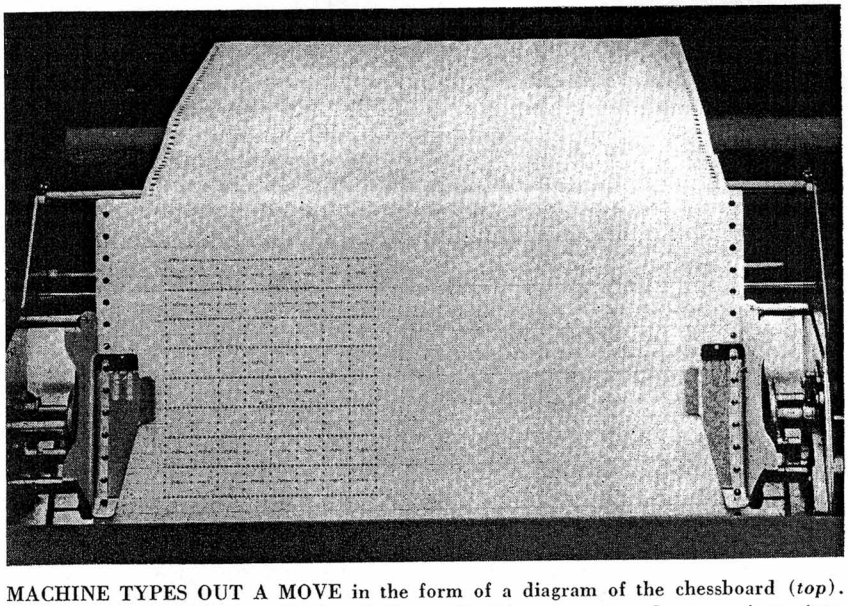
terms—that is, it must detect the strategy and anticipate the judgments of its human opponent. In other words, lacking the omniscience that would enable it to win no matter what its opponent does, it must try to outwit the opponent.

Needless to say, devising a program which would give a machine this property—what amounts to the capacity to think—has proved a very difficult job. The late A. M. Turing, the ingenious British theoretician on thinking machines, was one of the first to try his hand at designing a chess-playing program for a computer, but his machine (MADAM) played a very weak game, made stupid blunders and usually had to resign after a few moves. The problem has interested a number of computer experts in the U. S. [see "A Chess-Playing Machine," by Claude E. Shannon; SCIENTIFIC AMERICAN, February, 1950], and several groups are currently working on chess programs. We want to report here what we believe is the first satisfactory program—one with which the machine plays a game sophisticated enough so that its opponent has to be something more than a novice to beat it. The program was written by four collaborators—the authors of this article, who work for the International Business Machines Corporation, and Timothy Arbuckle and M. A. Belsky of the Service Bureau Corporation. It is designed for the IBM 704, the very rapid digital computer which has performed as many as one billion calculations in a single day in computing the orbit of an artificial satellite.

The program is a set of explicit instructions to the computer on how it must act in each of the specific situations with which it may be confronted. The instructions are given to the machine on a reel of magnetic tape. The operation of the computer is itself fascinating to watch. You sit at the console of the machine with a chessboard in front of you and press the start button. Within four seconds a panel light labeled "Program Stop" lights up on the console, and you now make your choice of black or white: to choose black you flip a switch on the console; if you want white, you simply leave the switch as it is. Suppose you have picked black. To begin the game you press the start button again. The machine now "thinks" about its first move. There is nothing spectacular about this. Some lights flash on the console, but the computer is working so swiftly that it is impossible to say just what these flashes mean. After about eight minutes, the computer







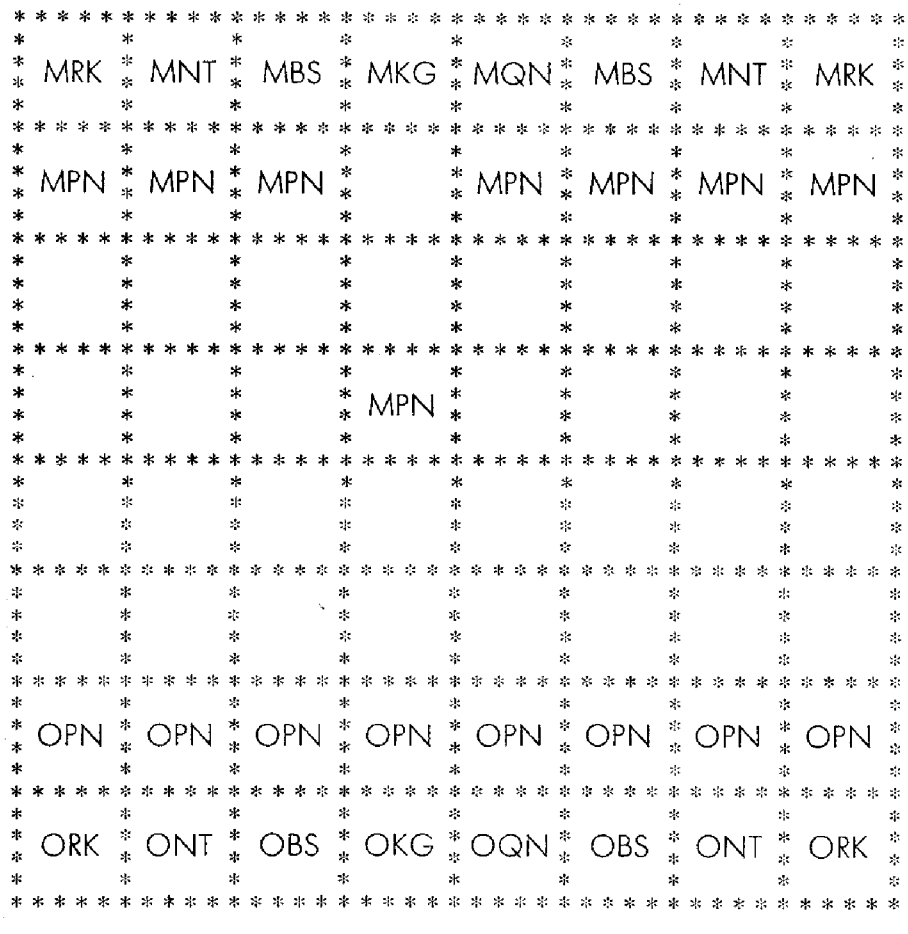
MACHINE TYPES OUT A MOVE in the form of a diagram of the chessboard (*top*). Bernstein makes the move on the board, then makes his own move and communicates it to the machine (*middle*). The machine types this move (*bottom*) before it makes its own.

prints out its move on a sheet of paper.

Let us say the machine's (White's) first move is king's pawn to the king's fourth square. The print-out then is W1 P-K4. The machine proceeds to print the chessboard with the positions of the pieces, designating its own by the letter M and its opponent's by the letter O [*see illustration below*].

Now the "Program Stop" light goes on again and the computer waits for its opponent to reply. You punch your replying move on an IBM card and put this card in a section of the machine which reads it. To signal that it is the machine's turn you press the start button again. The machine prints your move and the new board position and then goes on to calculate its second move. If you have made an illegal move, the computer will refuse to accept it, printing out "PLEASE CHECK LAST MOVE." So the game proceeds. At the end of the game, after a mating move or a resignation, the machine prints the score of the game, and to its opponent: "THANK YOU FOR AN INTERESTING GAME."

MACHINE




OPPONENT

CHESSBOARD TYPED OUT BY MACHINE represents the machine's pieces by M and the opponent's pieces by O. The second and third letters in each of the small squares represent rook (RK), knight (NT), bishop (BS), king (KG), queen (QN) and pawn (PN). In chess terminology the move shown here is P-K4 (pawn to king's column, fourth row).

In explaining the program of instructions to the machine it will be helpful if we start by contrasting it with an ordinary job performed by a computer—say calculating John Doe's pay check. The machine in the latter case simply takes the data—so many dollars for a 44-hour week, so much for overtime at a certain rate, so much deducted for social security and income tax—and quickly computes what the check has to be. There is one, and only one, correct answer. But in a chess game there are only two questions to which absolutely definite and unavoidable answers can be given: "Is this move legal?" and "Is the game over?" To all other questions there are various possible answers, though some may be more acceptable than others. The problem is to equip the machine with a system of evaluating the merits of the alternatives. This, as we have remarked, is what makes the task interesting. If cut-and-dried answers to all possible situations could be worked out by a computer, chess would immediately lose its fascination.

Obviously the machine's first job is to size up the board. The instructions therefore direct it to start by examining the state of the squares. The computer painstakingly and single-mindedly considers square by square, giving the same minute attention to squares of little interest as to those of key importance. It asks about each square whether it is occupied, by whose man, whether it is attacked, whether it is defended, whether it can be occupied. The information is summed up in tables compiled by the machine. All this takes about one tenth of a second, which is a long time by computer standards. The computer then proceeds to consider its best move.

Here we reach the most difficult and controversial part of the program, for to find a workable basis for the machine's decisions we must make some hypotheses about how a human being plays chess. To begin with, we have to decide on what basis a human player (or the machine) will select the moves that are to be given serious consideration (full consideration of all possible moves being out of the question). There are two distinct philosophies about this. One is that the player concentrates on the moves that look most plausible in the immediate situation. The other is that the player's approach to the selection is dictated by a grand strategy, and as far as he can he looks for moves which will further his plan. We built our program on the second hypothesis.

Of the various possible moves it might make (usually about 30) the machine selects seven for detailed analysis. It picks these on the basis of eight questions, which it asks in the following order:

1. Am I in check, and if so, can I capture the checking piece, interpose a piece or move away?

2. Are any exchanges possible, and if so, can I gain material by entering upon the exchange, or should I move my man away?

3. If I have not castled, can I do so now?

4. Can I develop a minor piece?

5. Can I occupy an open file?

6. Do I have any men that I can put on the critical squares created by pawn chains?

7. Can I make a pawn move?

8. Can I make a piece move?

Let us take the opening move for illustration. Examining the initial setup of the board, the machine finds that

questions 1, 2 and 3 must be answered "No." The answer to question 4 is "Yes"; the machine notes that it can move either knight and has four possible knight moves (N-KR3, N-QR3, N-KB3, N-QB3). To questions 5 and 6 the answer is "No." It is "Yes" to question 7. Any of the eight pawns may be moved, but the instructions tell the machine to give priority to P-K4, P-K3 and P-Q4. These three pawn moves, with the four knight moves, provide the machine with seven moves for study.

It now proceeds to test each of the seven in turn through four moves ahead, considering its opponent's possible replies and its own possible counter-responses in each case. The machine starts with one of the seven moves and asks itself what it might reply were it the opponent. It generates seven possible replies, on the basis of the questions listed above, and now it takes the first of these and considers its own possible responses. After generating seven plausible responses, it again takes the first of these and in turn generates seven plausible replies by the opponent to this move.

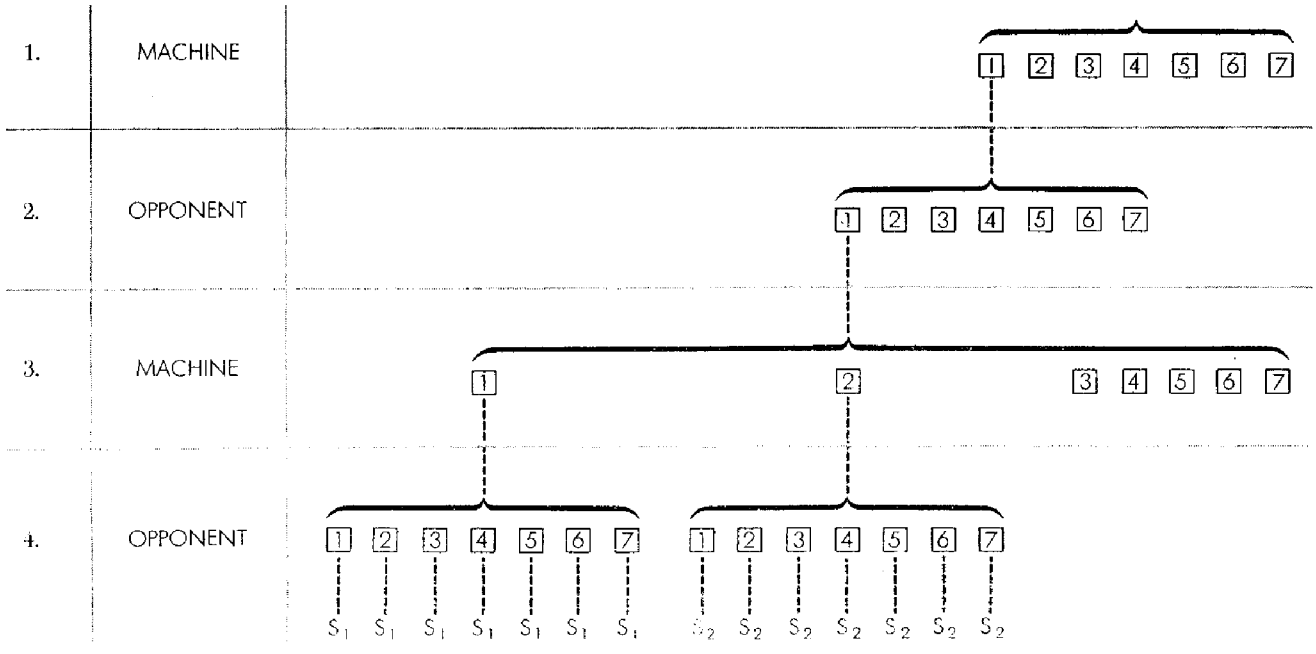
The machine has reached the fourth level: initial move, reply, counter-reply and now the opponent's seven potential responses to its counter-reply [*see diagram below*]. It goes on to examine each of these seven moves to see which one would net the highest value for its opponent. The value, or score, is measured by four considerations: (1) gain of material (a pawn counting as one unit, a knight or bishop three, a rook five and the queen nine); (2) defense of the king; (3) mobility of the pieces; (4) control of important squares. After the machine has determined the score for the opponent's best move in level 4, it carries this back as the score for its own move 1 in level 3.

In this manner the machine investigates all the possible sequences of plays, taking each of the seven moves at every level of the "tree," and arrives at scores for all the outcomes at the fourth level. In all, it examines 2,800 possible positions. After this examination, the machine then chooses as its first move the one that will lead to the highest score both for itself and for its opponent. It acts, in other words, as if its opponent will make his best possible moves within the limits it is programmed to explore.

These limits—four half-moves ahead with seven choices at each step—are dictated by the time factor. It takes the machine close to eight minutes to decide on each move in most cases. If it had to weigh eight plausible moves instead of seven at each level, it would take about 15 minutes for a move. If it carried the examination through to one more level ahead, a single move would take some six and a half hours. So the present program is considered about the limit for a machine operating at the speed of the IBM 704.

How does the machine make out with this program? In the first place, the machine is never absent-minded. It makes no blatant blunders such as letting a piece be caught *en prise*, as every chess master has done at some time or other. When its opponent is careless enough to expose a piece, the machine takes instant advantage of the opportunity to capture it. Secondly, in its choice of individual moves the machine often plays like a master, making what an expert would consider the only satisfactory move [*see example on page 6*]. Thirdly, the machine is certainly not in the master class in the play of a complete game.

A typical game played by the machine against a skillful opponent is illustrated on the next page. We have deliberately chosen a game which the machine lost, because we want to emphasize the point that a machine is not infallible and also because it is more instructive to watch the computer lose than to watch it win. The machine's opening moves in this game are quite acceptable. But by middle game the machine betrays its chief weakness: namely, a heavy bias toward




MACHINE MAKES A MOVE by the procedure suggested in this diagram. First, the machine selects, on the basis of eight questions, its seven most logical moves (*row 1*). Second, the machine selects its opponent's seven most logical responses to the first of these seven moves (*row 2*). Third, the machine selects its seven most logical counter-responses to the first of its opponent's responses (*row 3*). Fourth, the machine selects its opponent's seven most logical responses to the first of its seven counter-responses (*row 4*). Fifth, the machine scores its opponent's seven responses to the first of its seven counter-responses ($S_1$). Sixth, the machine selects its opponent's seven most logical responses to the second of its seven counter-responses. Seventh, the machine scores its opponent's responses to the second of its seven counter-responses ($S_2$). The machine continues in this manner until it has examined all moves.

| | MACHINE (WHITE) | OPPONENT (BLACK) |
|---|---|---|
| 1. | P — K4 | P — K4 |
| 2. | B — B4 | P — QN3 |
| 3. | P — Q3 | N — KB3 |
| 4. | B — KN5 | B — N2 |

Black is preparing for a direct attack on the center, via P — Q4.

| | | |
|---|---|---|
| 5. | B × N | Q × B |
| 6. | N — KB3 | P — B3 |
| 7. | O — O | P — Q4 |
| 8. | P × P | P × P |
| 9. | B — N5 ch | N — B3 |
| 10. | P — B4? | P × P |

White 10 N × P is better because if black replies Q × N, then R — K1. Since the pawn is defended by the queen, N × P seemingly loses material, and the move is discarded.

| | | |
|---|---|---|
| 11. | B × N ch | Q × B |
| 12. | P × P? | P — K5 |

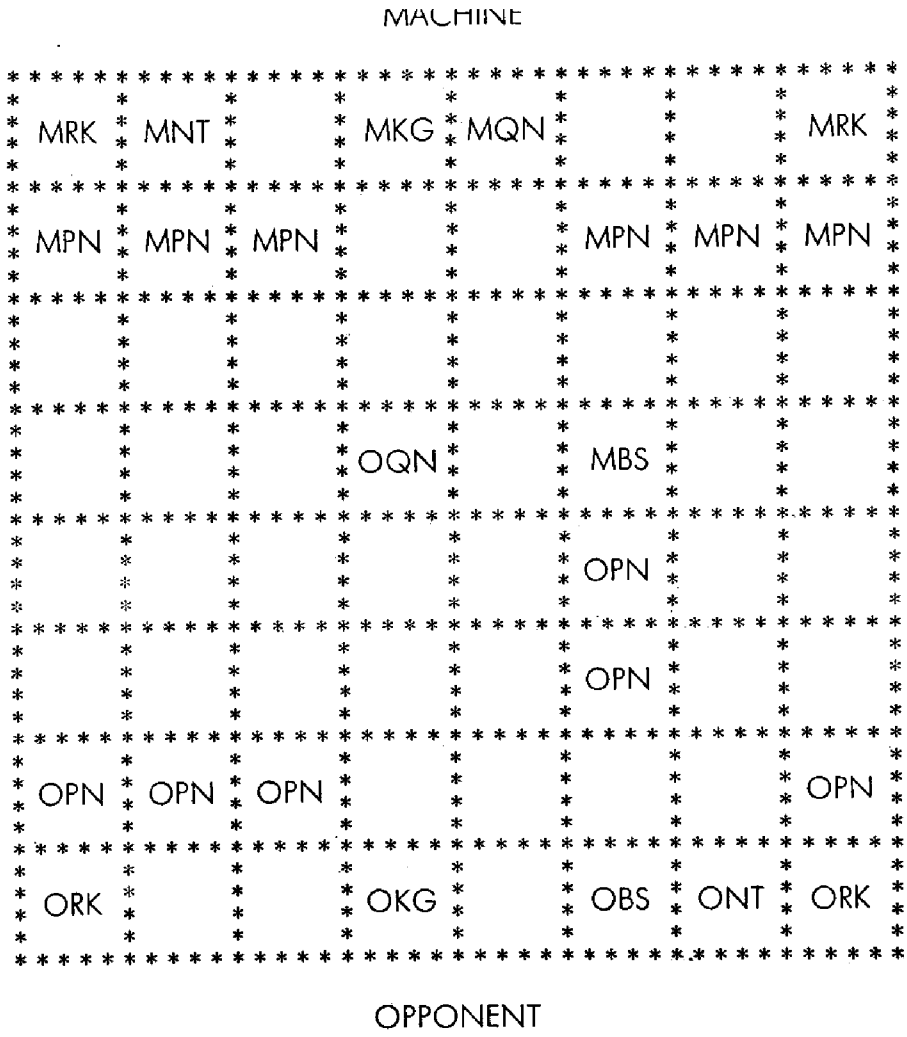
White 12 is bad, R — K1 is better.

| | | |
|---|---|---|
| 13. | N — N5 | Q — N3 |
| 14. | N — KR3 | P — K6 |
| 15. | P — B3 | B — B4 |
| 16. | R — K1 | O — O |
| 17. | N — B3 | |

Fiddling while Rome burns.

| | | |
|---|---|---|
| | | P — K7 dis ch |
| 18. | N — B2 | B × P |
| 19. | P — KN3 | P × Q = Q |
| 20. | N (QB3) × Q | Q — B7 |
| 21. | P — N3 | R (QR1) — Q1 |
| 22. | P — KR4 | R × N |
| 23. | Resigns | |

**ACTUAL GAME between computer and human opponent is described in conventional chess terminology. The comments of the human opponent have been interpolated.**




**MASTERLY MOVE was made out of this position by the machine. The move was Q-K2 (queen to king's column, row 2). Experts would consider this the only satisfactory move.**

moving attacked pieces rather than defending them (a weakness which could be corrected only by increasing the time for considering moves). At the tenth move White (the machine) makes a weak move which puts Black in a strong position; by the thirteenth move White's position is clearly hopeless, and 10 moves later, seeing the inevitability of a forced mate, the machine resigns.

Our contests with the machine show that anyone good enough to construct a three-move trap can beat it. Knowing how it selects its moves for consideration, you can often think of moves which you can be confident the machine will not consider. The machine will invariably accept a "sacrifice" (but then, so did the grand master José Capablanca). It will offer a sacrifice only to avoid being mated or if it can see an almost immediate mate of its opponent.

Yet notwithstanding its weaknesses, the IBM 704 plays a respectable and not-too-obvious game of chess—a game about which one can ask such questions as "Why did it make that move?" and "What does it have in mind?" We can even say frequently that "It made an excellent move at this point," or "At this stage it had a good position."

Undoubtedly our chess player is only a prototype for far more skillful players to be built in the future. Probably they will not go much farther in depth of planning: even with much faster computers than any now in existence it will be impracticable to consider more than about six half-moves ahead, investigating eight possible moves at each stage. A more promising line of attack is to program the computer to learn from experience. As things stand now, after losing a game the machine quite happily makes the same moves again and loses again in exactly the same way. But there are some glimmerings of ideas about how to program a machine to avoid repeating its mistakes, and some day—not overnight—we may have machines which will improve their game as they gain experience in play against their human opponents.