



Reconstructing Turing's "paper machine"

Garry Kasparov and Frederic Friedel

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 14, 2017

Reconstructing Turing's "paper machine"

By Frederic Friedel and Garry Kasparov

It is an amazing fact that the very first chess program in history was written a few years *before* computers had been invented. It was designed by a visionary man who knew that programmable computers were coming and that, once they were built, they would be able to play chess.



The man, of course, was Alan Turing (sculpture above in Bletchley Park), one of the greatest mathematicians who ever lived. He was very interested in chess, but in spite of having a brilliant intellect and putting a lot of effort into learning the game he remained a fairly weak player. While working at Bletchley Park on breaking the German Enigma code (and probably decisively influencing the outcome of the War) he took chess lessons from his GM-strength colleagues in the decryption team, and even invented "[Round-the-house chess](#)" (after each move you had to run around the house, and could play two moves in a row if you overtook your opponent). Turing was an Olympic-class runner, and this was his way of balancing out his lack of chess talent.

In 1948 Turing, assisted by his friend David Champernowne, wrote the instructions that would enable a future machine to play chess. They called the program *Turochamp*, but it popularly became known as "Turing's paper machine." Turing's goal was to "make a machine which would play a reasonable good game of chess, i.e. which, confronted with an ordinary chess position, would after two or three minutes of calculation, indicate a passably good legal move." The program introduced the concept of quiescence. Champernowne wrote: "Most of our attention went to deciding which moves were to be followed up. Captures had to be followed up at least to the point where no further capture was immediately possible. Check and forcing moves had to be followed further. We

were particularly keen on the idea that whereas certain moves would be scorned as pointless and pursued no further others would be followed quite a long way down certain paths.”

Turing (and Champernowne) used the following piece values: pawn=1, knight=3, bishop=3½, rook=5, queen=10. In addition they had the following positional evaluation functions:

(1) Mobility. For the Q,R,B,N, add the square root of the number of moves the piece can make; count each capture as two moves.

(2) Piece safety. For the R,B,N, add 1.0 point if it is defended, and 1.5 points if it is defended at least twice.

(3) King mobility. For the K, the same as (1) except for castling moves.

(4) King safety. For the K, deduct points for its vulnerability as follows: assume that a Queen of the same colour is on the King's square; calculate its mobility, and then subtract this value from the score.

(5) Castling. Add 1.0 point for the possibility of still being able to castle on a later move if a King or Rook move is being considered; add another point if castling can take place on the next move; finally add one more point for actually castling.

(6) Pawn credit. Add 0.2 point for each rank advanced, and 0.3 point for being defended by a non-Pawn.

(7) Mates and checks. Add 1.0 point for the threat of mate and 0.5 point for a check.

At the time there was no machine, of course, that could execute this set of instructions, and so Turing acted as a human CPU, using paper and pencil, requiring more than half an hour to calculate each move. He played a game against Champernowne's wife, who was a beginner at chess, and won. Then he played one against a colleague, Alick Glennie, which he lost. This second game was recorded:

Turing's paper machine – Alick Glennie, Manchester 1952: 1.e4 e5 2.Nc3 Nf6 3.d4 Bb4 4.Nf3 d6 5.Bd2 Nc6 6.d5 Nd4 7.h4 Bg4 8.a4 Nxf3+ 9.gxf3 Bh5 10.Bb5+ c6 11.dxc6 0-0 12.cxb7 Rb8 13.Ba6 Qa5 14.Qe2 Nd7 15.Rg1 Nc5 16.Rg5 Bg6 17.Bb5 Nxb7 18.0-0-0 Nc5 19.Bc6 Rfc8 20.Bd5 Bxc3 21.Bxc3 Qxa4 22.Kd2 Ne6 23.Rg4 Nd4 24.Qd3 Nb5 25.Bb3 Qa6 26.Bc4 Bh5 27.Rg3 Qa4 28.Bxb5 Qxb5 29.Qxd6 Rd8 0-1.



In the early 1950s Turing began programming Turochamp on a Ferranti Mark 1 computer (above, image courtesy of [Chess Programming Wiki](#)) at the University of Manchester. At the same time he tried implementing a rival program, *Machiavelli*, by Donald Michie and Shaun Wylie on the same machine. Unfortunately he had not completed the task before his tragic death in 1954.

Interestingly nobody seems to have written an actual computer program using Turing's algorithm for more than fifty years after Turing devised it. In 2004 the programming team at the ChessBase software company decided to create a chess engine based on Turing's algorithms. The basis was Turing's description and the sample game against Glennie, and the programmer who undertook the task was Mathias Feist. He built a standard chess program interface for the ChessBase *Fritz* program, so that the Turing engine could be tested.



In the process, however, the ChessBase team encountered a problem: the engine refused to duplicate all of Turing's moves as recorded in the Glennie game. It deviated in ten places – significantly already on the first move: the Turing engine wanted to play 1.e3, whereas Turing had executed 1.e4 in his 1952 game.

The ChessBase team spent some weeks re-examining the Turing papers and discussing the implementation in the Turing engine. Since they could not locate the error they consulted Ken Thompson, one of the pioneers of computer chess (and author of Unix and C). He spent some time wrestling with the problem, and in fact wrote his own code on the basis of his reading of the Turing instructions. His program produced most of the ChessBase deviations from the Glennie game. It was quite baffling.

So Frederic Friedel contacted Donald Michie, who had been with Turing at Bletchley, describing the problem in general and the most significant deviations the ChessBase Turing program produced after the prescribed three-ply search (which Turing used in his paper-and-pencil game). "It is possible that we have got something wrong," he wrote, "but I doubt it, since very often, especially in the beginning, we get the same moves with exactly the same values. I think it is possible that Turing tired after fifteen moves, when in addition the position became quite complex?!"

Michie's reaction (in a phone conversation) was: "You are looking for a bug in the program, Frederic? No, no, you should look for it in Alan Turing! Alan did not care about details; he was interested in the general principle." He also pointed us to a quote by Champernowne, who had assisted Turing during the 1952 game: "In the actual experiment I suspect we were a bit slapdash about all this and must have made a number of slips since the arithmetic was extremely tedious with pencil and paper."



In June 2012 Friedel travelled to Manchester to attend the Alan Turing Centenary Conference. On the way he spent June 23rd, Turing's 100th birthday, at Bletchley Park, a visit he will never forget. In Manchester he assisted Garry Kasparov in a commemorative lecture on Turing's chess involvement. The two spoke about the reconstruction of the paper machine, and [Kasparov actually played a game](#) against it. You can watch the entire one-hour lecture in the video given below.

In the lecture Kasparov pointed to a key moment in the game: "It was the first time that a machine played a game of chess... After a pretty average game they reached an equal position and the machine blundered a queen – taking the pawn, after which the queen is pinned, and White resigned."



Kasparov: “Blundering a queen in one move, that is not a great accomplishment. But still the game was played and there were 29 moves, not just legitimate moves but you may call them decent moves.” He confirms that he will play a game against the reconstructed engine at the end of the lecture, “the first official display of the Turing machine playing against a professional player. ... I think that at five seconds a move it could beat most amateurs...”

In the lecture Friedel discussed the deviations we were confronted with and how they probably came about. An important point is that Turing, working with paper and pencil, clearly used his intuition to come up with the moves. And without knowing it he used Alpha-Beta pruning, a technique that was invented by Allen Newell and Herbert A. Simon half a decade later, when machines could actually execute code. Essentially Turing did not bother to calculate every single move in a position. Some he thought were clearly inferior, and it seemed pointless to spend a lot of time working out exactly how bad they were (which is the point of Alpha-Beta).

In the lecture we discussed the problem of the very first move: Turing played 1.e4, the ChessBase engine (and Thomson’s reconstruction) come up with 1.e3.



In the initial position material is equal, no pieces are taken, nothing has happened yet. There are twenty legal moves for White – each of the eight pawns can move up one or two squares, and the two knights have two moves each. One wonders if Turing made 20 calculations, each taking a number of minutes, before he executed the first move, 1.e2-e4. It is one of the two most common moves that start a chess game (the other is 1.d2-d4).

Let us apply Turing’s rules to the initial position. The e-pawn gets 0.4 points for moving two squares forward, but just 0.2 points for moving one step (rule 6: pawn credit). So 1.e4 is better. But then we apply rule 4 (king safety) by placing a queen on the square e1, counting the number of squares the

queen can move to – they are the squares from which the king could be attacked by a rook or a bishop – and taking the square root to define its mobility. 1.e2-e3 gets a malus of 1 (square root of one square, e2, to which the queen can move), whereas 1.e2-e4 reduces the positional advantage by 1.4 points (square root of 2, for e2 and e3).

In his game score Turing gives 4.2 points for the move he played, 1.e2-e4, and adds an asterisk to the value, which indicates that “every other move has a lower position-play value.” The ChessBase engine, and that of Ken Thompson, give 1.e2-e3 a positional value of 4.40, compared to Turing’s move: 0.2 less for advancing just one pawn but 0.4 more for king safety. Mathias Feist wrote “Turing’s calculation for e2-e4 is correct, which means he did not really calculate e2-e3. As a chess player he knew that this move is not logical, because in the initial position king safety doesn’t play a role. He probably looked at it and thought: everything is the same for both moves, except e2-e3 is 0.2 points worse because the pawn moves just one square. So he discarded it without further calculation.”

Incidentally the move 1.d2-d4, one of the most popular in tournament chess, was certainly calculated by Turing, who must have seen that it is clearly worse. Putting a queen on the king square shows four squares to which it can move, i.e. the malus for king safety is 2.0.

Let us take a look at move four: Turing played 4.Ng1-f3 and gave a value of 2.0. The ChessBase Turing engine plays 4.d4xe5 and give it a value of 1.1. Why?



After 4.Ng1-f3 the mobility of the queen on d1 sinks from six to three squares, and consequently the positional value from 2.4 to 1.7 (= -0.7). The mobility of the knight g1 increases from three to five squares, and so the score increases from 1.7 to 2.2 (= +0.5). The mobility of the rook on h1 increases

from 0 to 1 square, improving the score from 0.0 to 1.0 (= +1.0). So 4.Ng1-f3 improves the evaluation of the position by 0.8 points.

After 4.d4xe5 the d4 pawn has advanced by one square (+0.2) but is no longer defended (-0.3). The mobility of the queen increases from 6 to 9 squares (+0.6). The black pawn on e5 has disappeared, and with it the bonus for the two rows it had advanced (+0.4 for White). The black king safety increases from 4 to 5 (+0.2 for White). So in total 4.d4xe5 increases the score by 1.1 points.

There were also some problems with the quiescence search (“captures had to be followed up at least to the point where no further capture was immediately possible”), as applied by Turing in the Glennie game.



On move five Turing played Bc1-d2 and gave a value of 3.5. But after 5.Bd2 exd4 6.Nxd4 Bxc3 7.bxc3 (or Bxc3) Nxe4 no more captures are possible and White has lost a pawn, while after 5.Bg5 exd4 6.Nxd4 Bxc3 7.bxc3 (or Bxc3) Nxe4 White plays 8.Bxd8 Kxd8 and in the resulting quiescent position White has a queen (10 points) for a knight and pawn (3+1).



Similarly in the above position Turing played 15.Rh1-g1 and gave it a 1.2 score. But the quiescent search reveals that 15.Rg1 Bxf3 16.Qxf3 Qxa6 loses a pawn for White, which 15.Rh1-h3 does not.

There are a number of other deviations in the game, more towards the end, when Turing was apparently tiring. The game clearly demonstrates how tedious and error-prone human calculation in such a situation can be. On a modern computer the ChessBase Turing engine needs less than five milliseconds to calculate the values for all legal moves in a given position, and to choose the best one to play. Turing took fifteen to twenty minutes.

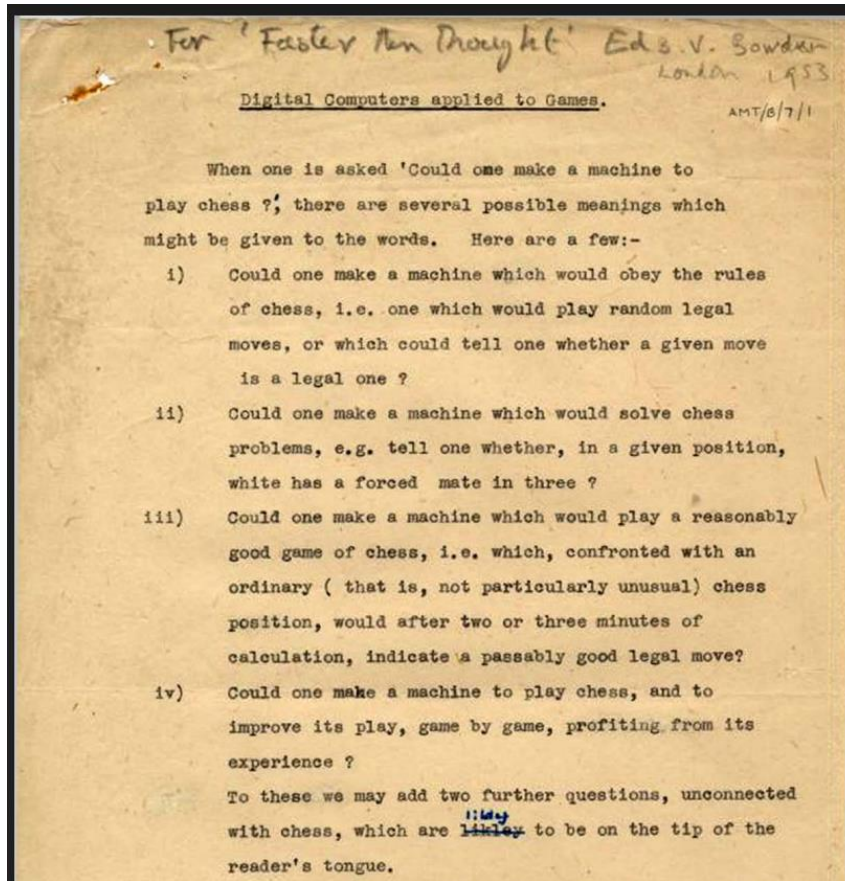
For those of you who want to experiment with the ChessBase Turing engine – or simply play a game against this historical chess program – [we have a download](#) you can install on your computer. It will run as a UCI engine on any Fritz-compatible chess software.

The Reconstruction of Turing's "Paper Machine"

[Video lecture by Garry Kasparov and Frederic Friedel](#)

References

- Monroe Newborn: Computer Chess. Academic Press 1975, pp.15-18
- Alex G. Bell: The Machine Plays Chess? Pergamon 1978, pp.12-23
- David Levy & Monty Newborn: How Computers Play Chess. W.H. Freeman 1991, pp.34-38
- David Levy: Computer Chess Compendium. B.T. Batsford 1988. This collection gives a standard reprint of Turing's article "Chess", first published in *Faster Than Thought* (B.V. Bowen, Editor) London, Pitman 1953, pp.286-295



There is a scan of the Faster-Than-Thought paper [on Google Docs](#). It is the original submission by Turing, typed on yellow paper, with ink corrections. Turing quotes the game against Glennie, but deviates on move 21 from the gamescore that was later published in all the reprints. It is to be assumed that he corrected this first version of his manuscript with the moves that have been preserved for posterity.