

The Chess Example in Turing's Mind Paper is Really about Ambiguity

Jeroen Fokker

Dept. Computer Science, Utrecht University, The Netherlands
J.D.Fokker@uu.nl

Abstract

In his paper *Computing machinery and intelligence*, Turing introduces a rather trivial chess problem as a conversation piece in an example Turing test. The example is about much more than computers playing chess: it has a hidden message that is about resolving ambiguity and knowledge representation, probably inserted by Turing as an Easter egg for us to find.

1 Introduction

Q: Do you play chess?
A: Yes.

This question is one of the examples that Turing gives while explaining an interrogation game that came to be known as the ‘Turing test’ [5]. In the game, an interrogator can ask questions through a chat session, and has to tell from the answers whether the conversation partner is human or machine. The test is introduced in an operationalization of the question ‘Can machines think?’. A machine that can’t be discerned from a human in such an interrogation would be regarded as ‘intelligent’.

Turing argues that this setting is ‘suitable for introducing almost any one of the fields of human endeavour that we wish to include’. The examples that Turing gives show the tactiques that the machine could employ to convince the interrogator that it actually is human:

Q: Please write me a sonnet on the subject of the Forth Bridge.
A: Count me out on this one. I never could write poetry.
Q: Add 34957 to 70764.
A: (Pause about 30 seconds and then give as answer) 105621.

Turing does not explain the examples, but from the poetry example the reader can learn that it is possible to refuse to enter some areas of discourse. The arithmetic example shows that it would be wise for a machine not to calculate too quickly, as that might reveal its mechanic nature.

There is a more subtle point in the arithmetic example, often discussed in commentaries: the answer is wrong! (The correct answer being 105721). This hidden joke can be considered as an Easter egg for the (very) alert reader. On the surface, it can be understood again as a wise tactique for the machine to throw in an error now and then, just to look human. Considering again, we realize that this is exactly the kind of errors that a human calculator is likely to make: missing a carry in the 100’s place, where there a confusingly many carries anyway.

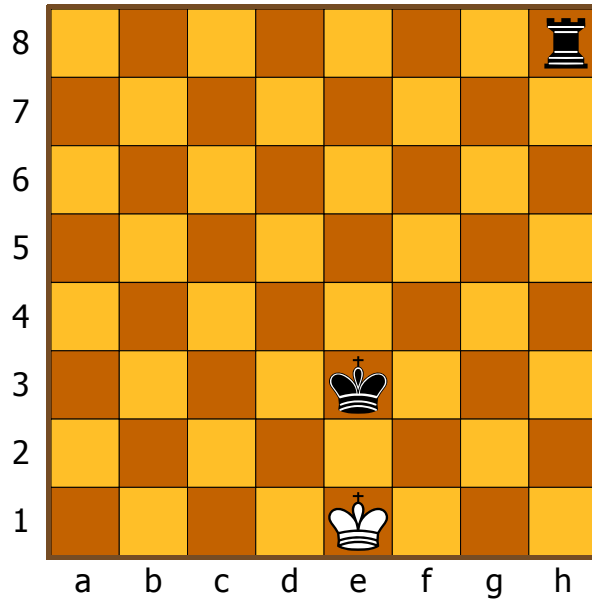


Figure 1: The chess example in Turing's paper

In a commentary, Douglas Hofstadter even goes as far as considering the possibility that the mistake may be *accidentally* made by a machine that is really emulating the human way of doing arithmetic, and conjectures that this philosophical question was raised intentionally by Turing in the Easter egg [3].

Turing's example conversation continues with two questions on chess:

Q: Do you play chess?

A: Yes.

Q: I have K at my K1, and no other pieces. You have only K at K6 and R at R1.

It is your move. What do you play?

A: (After a pause of 15 seconds) R-R8 mate.

On the surface, it can be understood as a requirement that a candidate intelligent machine should be able to play chess (not entirely obvious when the paper was written in 1950), or at least to solve a simple mate-in-1 problem (see figure 1). This example, just as the arithmetic example, has a hidden layer as well, that is never mentioned in commentaries. We will discuss in this paper that it is not (only) about chess, but also about language, resolving ambiguity, and knowledge representation.

2 A Problem with Notation

The chess problem given by Turing is a simple mate-in-1 problem. For the modern reader this may not be obvious without figure 1, as the problem is denoted in the so-called *descriptive* or

8	TR8	TN8	TB8	TQ8	TK8	TB8	TN8	TR8
7	QR7	QN7	QB7	Q7	K7	KB7	KN7	KR7
6	QR6	QN6	QB6	Q6	K6	KB6	KN6	KR6
5	QR5	QN5	QB5	Q5	K5	KB5	KN5	KR5
4	QR4	QN4	QB4	Q4	K4	KB4	KN4	KR4
3	QR3	QN3	QB3	Q3	K3	KB3	KN3	KR3
2	QR2	QN2	QB2	Q2	K2	KB2	KN2	KR2
1	QR1	QN1	QB1	Q1	K1	KB1	KN1	KR1
	a	b	c	d	e	f	g	h

Figure 2: Numbering of chess board squares in the English notation, for the white player shown white upright, and for the black player shown black and upside down

English notation. It was widely used in 1950, especially in Britain, but is now replaced by the standard *algebraic* notation.

In the algebraic notation, the files (columns) of the board are designated by letters a–h, and the ranks (rows) are designated by numbers 1–8. Pieces are indicated by capital abbreviation of their name (e.g. K for King or Q for Queen), or by a *figurine* like ♔ or ♚.

In the English notation, the files are designated by an abbreviation of the piece that occupies the first row in the initial position, e.g. Q (Queen) for the d-file, and K (King) for the e-file. Thus, ‘I have K at my K1’ is represented in algebraic notation as Ke1 (assuming that the interrogator plays white).

Furthermore, in the English notation, positions are indicated *relative* to the player. Thus, when ‘you have K at K6’, the black player has Ke3, not Ke6.

Since each player has two bishops (B), two knights (N), and two rooks (R), both the c-file and the f-file are designated B in the English notation. To disambiguate, it should be prefixed with Q or K, to indicate the queen’s or king’s side of the board. Thus QB1 is c1 (or c8 for the black player), and KB1 is f1 or f8. Likewise, QR1 is a1 or a8, and KR1 is h1 or h8. Refer to figure 2 for all 64 square designations.

Now in Turing’s conversation, ‘you have only K at K6 and R at R1’ is ambiguous: it does not specify whether the rook is at QR1 or at KR1, that is a8 or h8. Thus a natural answer to the given problem would be: ‘what do you mean: is the rook at QR1 or KR1?’. Yet the example interrogee does not, and happily solves the problem in 15 seconds. Which of the two possibilities did he/she (or it) choose?

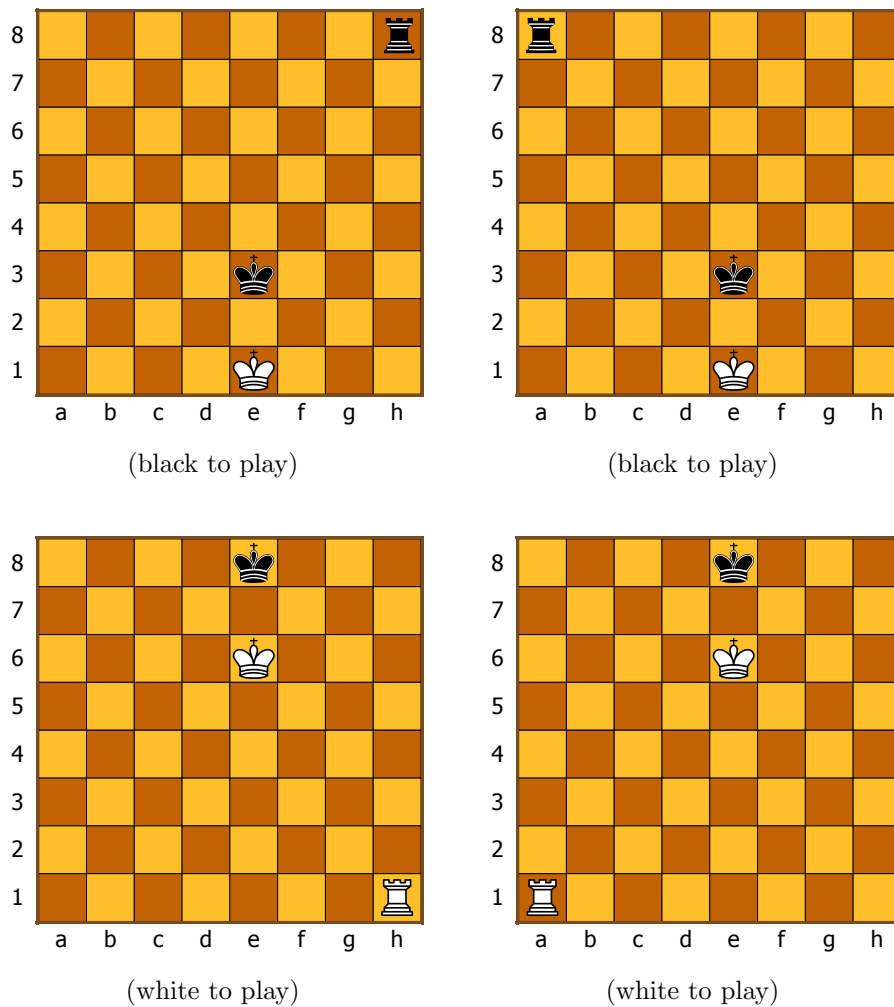


Figure 3: Four possible interpretations of Turing's chess problem

Actually, it solved both, because the answer 'R-R8 mate' is just as ambiguous as the question: it is the English notation for 'move the R piece to the R8 square', again not specifying whether QR8 or KR8 is meant. So, in the 15 seconds the interrogee must have found out that the problem is ambiguous, solved both versions, and realized that the respective solutions (R-QR8 and R-KR8) could both be abbreviated to R-R8, which is therefore a correct answer for both interpretations of the problem. That answer can thus be given without a need for further clarification.

And wait! We have made another assumption, namely that 'I' play white and 'you' play black. The problem as stated does not specify this. It could just as well be the other way around. 'I have K at my K1' then means that I have a black king at e8, and 'your' white king is at e6 and the rook is at either a1 or h1. Again there are two solutions, but due to the relative nature of the British notation, they can both be denoted as 'R-R8'.

3 Discussion

The possibility of a computer playing chess was highly intriguing in 1950. It was clear to Turing that a machine could be programmed to make legal moves, but being able to play chess, let alone at world top level, was long considered to be the exclusive realm of human intelligence. An example involving chess therefore makes sense in a 1950 paper on artificial intelligence.

But why did Turing choose such a trivial problem, that even in 1950 would have been hardly a challenge to program? Maybe it was just for brevity, not to clutter up the paper with chess technical detail. However, considering the richness of the rest of the paper, it is unlikely that the example was carelessly chosen. The rather unusual phrasing (involving 'I' and 'you' rather than just 'white' and 'black') suggests that the 4-fold ambiguity was introduced intentionally by Turing. The presence of an Easter egg in the arithmetic example increases the likelihood that there may be more issues raised without being discussed explicitly.

Indeed, the mere parsing of the questions is far more difficult to program than the actual chess problem solver. The interrogee has to recognize that chess jargon is used ('K at K6'), although it is mixed with common language ('K at *my* K1', 'no other pieces'). Thus, this single example touches many of the problems in natural language processing and discourse representation that are ubiquitous for participants in the Turing test.

The abilities needed to recognize that the problem is ambiguous, but that the ambiguity is not worth complaining about if the answer is given ambiguously as well, go beyond mere language processing: they require reflection on the problem solving process. That activity indeed is typical for intelligence.

A computer program that has to fall back on a chess playing subroutine employing a 2D array as board representation probably needs to go through all these steps. A human chess player may alternatively represent the chess problem at hand as an abstract relationship between the two kings and the rook, and think of the solution as 'moving the rook along the file'. (Indeed, I only became consciously aware of the ambiguity when I tried to translate Turing's problem into algebraic notation for a present-day non-British audience). Programming a computer to represent a chess situation in such conceptual terms instead of an 8×8 array would be a radical different approach, and is as hard to imagine as Hofstadter's calculator which *really* 'thinks' that 105621 is the correct sum, instead of cleverly trying to fool a judge by throwing in some random errors.

4 Related and Future Work

The commentaries on Turing's paper are countless. Most often, they discuss the philosophical implications, the validity of the Turing test as operationalization of intelligence, etc. Robert Epstein et al. recently edited a book giving a comprehensive overview of the state of affairs [1]. The chapter by Robert Horn [4] in particular charts 800 different views in the debate initiated by Turing.

A more practical approach is a competition of computer programs actually participating in a Turing test, known as the Loebner competition. Concrete tactics to fool a judge become more important here than the quest for intelligence. Kevin Warwick recently gave an overview of issues raised in connection to Loebner tournaments [6].

Fewer papers comment on the actual examples in Turing's paper; where they do so, this is



mostly about the arithmetic example. Only Mark Halpern [2] comments specifically on the chess example, and gives another interesting observation. The interrogator just asks 'What do you play?', and it is tacitly assumed that the interrogee chooses a winning move. The fact that one plays to win is common sense knowledge that is hard to program.

As for future work: given that both the arithmetic example and the chess example contain hidden layers, one might wonder whether there is a hidden layer as well in the third example question that Turing employs:

Q: Please write me a sonnet on the subject of the Forth Bridge.

What is so special about the Forth Bridge (a famous 19th C. railway bridge over the Firth of Forth near Edinburgh, Scotland)? Maybe that its name can be misunderstood as 4th bridge? Shakespearean connotations? Count me out on this one. I never could write on poetry.

References

- [1] Robert Epstein, Gary Roberts, and Grace Beber (eds.). *Parsing the Turing test: Philosophical and methodological issues in the quest for the thinking computer*.  Springer, 2009.
- [2] Mark Halpern. The trouble with the Turing test. *The New Atlantis*, 11:42–63, 2006.
- [3] Douglas Hofstadter. *Gödel, Escher, Bach: an eternal golden braid*. Basic Books, 1979.
- [4] Robert E. Horn. Mapping and navigating the debate. In *Parsing the Turing test* [1], pages 73–88.
- [5] Alan Turing. Computing machinery and intelligence. *Mind*, 50:433–460, 1950.
- [6] Kevin Warwick. Not another look at the Turing test! In *SOFSEM 2012: Theory and practice of computer science*, volume 7147 of *LNCS*, pages 130–140.  Springer, 2012.