Mechanizing Chess Games, Computable Enumerability and Dynamical Systems

K. Vela Velupillai

June 2020
Mechanizing Chess Games, Computable Enumerability and Dynamical Systems

K. Vela Velupillai‡
Tottvägen 11
169 54 Solna
Sweden

20th June, 2020

kvelupillai@gmail.com

‡ This is a Turing-Post centered piece. In this sense it is a companion piece to Velupillai (2013, 2017a, 2017b, 2020). As far as Simon is concerned, Velupillai (2018) is relevant, especially for §2, but it is my almost my first foray into anything by Shannon (except some notes on his work on the Differential Analyzer – which is viewed from the point of view of analogue computing): see also the interpretation in chapter 4 of Simon, *ibid*, of his early work on cybernetics and Velupillai, 2004).
Abstract

In this paper, the historical lines of research by Alan Turing, Claude Shannon and Herbert Simon, are explored, via their pioneering research on the possibilities of Computer Chess. Mechanizing chess and the resulting computably enumerable sets and the basins of attraction of viewing chess moves as dynamical systems, in finite game playing, makes it possible for machine intelligence and learnability to be analysed effectively.

Keywords: Chess, Computably enumerable sets, dynamical systems, Busy Beaver Game, machines, algorithms
§ 1. Introductory Speculations

“The game of Chess is not merely an idle Amusement. Several very valuable qualities of the mind, useful in the course of human Life, are to be acquired or strengthened by it, so as to become habits, ready on all occasions. For Life is a kind of Chess . . . . .”
Franklin, 2004, p.317 [before 28 June 1779]; capitals in original, italics added.

“You do not learn about your opponent’s character when you play Go or when you play chess. … Trying to judge the opponent’s character perverts the whole spirit of the game.”
Kawabata. 1972 [1951], p. 79; italics added.

Just in the XX century, Munshi Premchand¹ and Yasunari Kawabata in literature, Leo Tolstoy and Lewis Carroll as authors of different genres, Marcel Duchamp and Gabriel Orozco in art², Satyajit Ray and Ennio Morricone³ in films and music, the Dalai Lama and Pope John Paul II (Karol Wojtyla) representing a version of Buddhism and the Catholic Church, Vladimir Lenin⁴ and Mao Zedong⁵, the one a political revolutionary and the other as a guerrilla warfare strategist (and, of course, a political revolutionary), Charlie Chaplin and Humphrey Bogart as famous actors, Albert Einstein and Richard Feynman as mathematical physicists, Edmund Landau and Luitzen Brouwer⁶ as mathematicians are some of the luminaries who were passionate about Chess, Shogi (Japanese Chess) or GO.

¹ Premchand’s 1924 short-story, The Chess Players (Shatranj Ke Khilari) was made into a film (of the same name) by Satyajit Ray, in 1977.
² See Flanagan (2009) on Duchamp and Orozco and Kuenzli & Naumann (1990, edited) for Duchamp, alone; I hope Gabriel Orozco is not ‘confused’ with José Clemente Orozco!
³ Alessandro De Rosa (2019, editor).
⁴ There is an alleged etching by the young Hitler’s art teacher, Emma Lowenstramm, showing Lenin playing chess against the budding Nazi (The Telegraph, 3 September, 2009)
⁵ Boorman (1969).
⁶ van Dalen (2013), p. 643, where Brouwer is described as being ‘a fervent chess player’ and that this may be the reason for Max Euwe (who defeated Alekhine for the World Chess championship in 1935) being his PhD student. Incidentally, many seem to attribute to von Neumann the ‘first’ devising, in 1928, of the min-max (or saddle-point) property of a two-person game equilibrium concept; Euwe, working within Brouwer’s set theory, developed a constructive min-max solution, also in 1928.
I feel it is possible to interpret their professional endeavours in terms of the deep\(^7\) interests in one or other of these board games. This makes it possible to make sense of Kawabata’s observation about the futility of trying to understand the *character* of an opponent. After all, machines – whether intelligently learning or not – do not try to ‘judge the opponent’s character.’ Machines, and humans, seem to *know* (in epistemological ways) the futility of trying to define *character* and still maintain the *fiction* that the board is made up of a *finite* number of squares – whether the game being played is chess, shogi or GO.

\(^7\) The word *deep* is used in the way moves and strategies of play are evaluated in chess and GO by humans and machines (the latter implementing various optimizing techniques of learning, and searching along, tree strategies, by using neural net structures).
In computer (board) games, machines \textit{play to win}, in whatsoever way \textit{winning play} is defined. After all, the classic volumes of Berlekamp \textit{et. al.}, (1982) is titled \textit{Winning Ways}.

‘The world of Go’, as a board game, may ‘have its conscience and its ethics’ (Kawabata, \textit{op. cit.}, p. 47; italics added), if so, the machine must have them, too.

The paper is structured as follows. In the next section there are some remarks on the pioneers of (digital) machine computations, particularly on playing chess games. In §3, chess play is interpreted as finite Lachlan games. §4 is devoted to interpreting moves in chess as dynamical systems. The concluding §5 is mainly a plea for a purely machine defined conceptual space, in which humans learn and show intelligent behaviour (whether in board games or life, in general).

\section*{§ 2. The Pioneers – Turing, Shannon & Simon}$^8$

\textquote{Game playing was an early domain of interest, and Shannon, Turing, and Newell, Shaw and Simon contributed classic analysis of how machines might be programmed to play chess.”} Michie, 1986, p. 133; italics added$^9$.

I shall not unnecessarily duplicate what is widely available, in numerous articles and books, on the remarkable distinctive contributions to digital computer-based chess machines made by Alan Turing, Claude Shannon and Herbert Simon; of the many books and articles available for any interested person to consult, I myself have found the two edited volumes of Levy (1988a, 1988b), Newborn (1997, especially chapter 2), but above all Hodges (1983, p. 211, ff) and Newell & Simon (1972, chapter 4), most useful (obviously also the Turing, Shannon, Newell and Newell \textit{et.al.}, chapters in Levy’s books).

I would like to concentrate on a few of the remarkable insights that these pioneers have left as their respective legacies to digital computer-based chess playing machines. Michie (1986, particularly chapters 1 & 2) are relevant for understanding the pioneers’ work on games in general, but chess machines and their algorithms, in particular, using trial and error and distinguishing puzzle vs. game learning, in finite processes.

$^8$ There were many others who contributed to the digital computer-based construction of game playing machines; but I choose to concentrate on these three outstanding scholars, partly due to competence, but mainly because I am particularly interested in their oeuvre with regard to machine methodology, epistemology and philosophy.

In one of the early modern forays into computer chess, Alan Turing maintained that machines ‘must be allowed to have contact with human beings’ in actual situations. Adaptation is learning by intelligent machines and the moves in a finite game of chess provides the means to learn intelligently by machines. Turing, however, began to think of digital computer-based chess playing machines at least six years earlier (Hodges, op.cit., p. 211-212), whilst in full flow at Bletchley Park.

However, even Turing seemed to have been mildly prejudiced in favour of human beings! He wants machine to have contact with human beings, but not the other way about: human beings to have contact with machines so as to learn from them. I am sure he – as well as Shannon and Simon - would have applauded the feats of AlphaGoZero (Singh, 2017, Silver, et. al., 2017), but it must be remembered that it is a purely practical solution (to which the three pioneers were not averse).

In any case, in these observations by Turing (i.e., the quote above)\(^\text{11}\) he distinguishes between a machine and a human being; machines are not surrogate human beings. Secondly, he does not consider the Kasparov alternative (Sadler & Regan, 2019, p.8) of machine versus human beings, as against machine plus human beings\(^\text{12}\). Thirdly, the set of moves, of the machine and its opponent (either a machine or a human being), form a dynamical system. Since chess is a finite game of perfect information, with moves alternating between the two players, subject to the first move being decided by some rule (tossing a fair coin, for example), it is amenable to theoretical treatment – without any axiom – as a Busy Beaver Game (Rado, 1962). Therefore, Rado’s perceptive observation is relevant here (ibid, p.884; bold italics added):

“\[W\]e used in our constructions only the following ‘principle of the largest element’: If \(E\) is a non-empty, finite set of non-empty integers, then \(E\) has a largest element. … Our examples …”

\(^\text{10}\) All references to Turing (1992) include, also, the Notes to the respective chapters, of pp. 205-216.

\(^\text{11}\) I would say that they are also Shannon’s views; I am less sure about Simon!

\(^\text{12}\) I am tempted to think that it is a case of exclusive OR (Boole), as against the inclusive OR (Jevons); versus interpreted as the former and plus in terms of the latter.
show that this principle, even if applied only to exceptionally well-defined sets \(E\), may take us beyond the realm of constructive mathematics.”

That machine behaviour ‘may take us beyond the realm of constructive mathematics’ is what Turing means – I think – by ‘it [the machine] may’ have to adapt to the standards of human beings. For human behaviour, leading to moves in a game of chess can, at most, be within the realm of constructive mathematics.

There are three insights associated with Turing that are not generally mentioned in the context of digital computer-based chess playing machines\(^{13}\). First of all, Turing’s construction of chess machines assumed the practical workability Hilbert’s programme in strictly finite cases; hence, for example, the utilization of the tertium non datur and proof by contradiction was allowed in studying the behavior of chess machines. Chess was, for Turing, a finite game. Secondly, building a digital computer-based chess machine was similar to the construction of Bombes to decipher the German Enigma machine; it was a mathematical exercise based on his two fundamental contributions, the law of large numbers and the Turing Machine. Thirdly, constructing a digital computer-based chess machine was a stepping-stone towards machine intelligence\(^{14}\) and learning machines.

Turing (1992; [1948]) does allow unorganized machines to be organized as universal machines by means of character-expressions (and situation-expressions, ibid., p 121). These expressions can make finite-machines into infinite (countable or uncountable) machines, especially because ‘character may be subject to some random variation’ (ibid, italics added). This can be considered a fourth Turing insight and towards a theory of applying neural network theory and algorithmic information theory in the construction of machines that can play a game of chess intelligently and learning in the process. For this we must go beyond the mathematics of the traditional Turing Machine and have recourse to the Oracle Machines (or relative computation), whilst assuming that Chess is a (countably) infinite game (Copeland, 1999).

\(^{13}\) Machines, for Turing, Shannon and Simon, did not mean mechanical; there machine could be electrical, electronic, even hydraulic and, of course, hybrid (also in being partly digital and partly analogue).

\(^{14}\) Later, from the time of the Dartmouth conference (1956), machine intelligence came to be referred to as artificial intelligence by John McCarthy, Herbert Simon, Marvin Minsky and others.
Figure 4: In the summer of 1952, Turing played a match against Alick Glennie, a British computer specialist. Turing played the (digital) computer-based *Turochamp*\(^{15}\) program losing to Glennie in 29 moves. The board shows the positions of the pieces after the 29th move (see also Hodges, 1983, pp. 478-9).


**Claude Shannon**

“The investigation of the *chess playing problem* is intended to develop techniques that can be used for more *practical* applications.”

Shannon, 1950, p. 657; italics added.

It must be remembered that Shannon was a Professor of Communications *Science* (1957-58) before becoming a Donner Professor of *Science* at MIT for over twenty years (1958-1979). He was, essentially, an applied mathematician in the classic sense, not just a non-pure mathematician (although he had been a research assistant in mathematics in 1938-1940).

He met Alan Turing for the first (and ‘only’?) time in 1943 (Giannini & Bowen, 2017). As Copeland & Prinz (2017) surmise (p. 344; italics added):

“*Possibly* [Turing] *told* Shannon about the *computational chess* ideas that he {i.e., Turing} had previously discussed with Good and Michie, though no records exist of Turing’s conversations with Shannon, so we shall *never* know for sure.”

Shannon shared with Turing many of the personal traits that characterised them – both positively and negatively, for example both appreciated ‘ingenious machines and gadgets’ (Ioan, 2009, pp. 262), Shannon emphasizing the *engineering* aspects and Turing the mathematical and logical ones. Ioan went on (*ibid*.; italics added):

---

\(^{15}\) Alan Turing developed the (digital) computer-based program *Turochamp* with his friend, David Champernowne, in 1948.
“Chess-playing machines fascinated him; as early as 1950 he wrote a paper on *programming a computer to play chess*.\(^{16}\)

![Figure 5: Claude Shannon & Edward Lasker with Shannon’s relay-based Chess Machine, ca. 1950.](image)

*Gift of Monroe Newborn to the Computer History Museum*

(downloaded from Google article on Claude Shannon, on 7/4/2020)

Their fundamental disdain for convention, even as pure or applied mathematicians, but with immense respect for the former as providing foundations for the latter, is best described by Hodges (2008, p.4)\(^{17}\):

‘It was typical for [Turing] … to seek to outdo Bell Telephone Laboratories with his single brain and to build a better system with his own hands.’

As Hodges perceptively remarks (1983, p.61; italics added):

\(^{16}\) Shannon presented the paper on *Programming a Computer for Playing Chess* on March 9, 1949 (see footnote on p.637 of Shannon (1949) and Levy (1998a), p.1. However, he published it only in 1950 (Levy, 1998a, chapter 1.1; see also Copeland & Prinz, *ibid*).

\(^{17}\) As Hodges perceptively remarks (1983, p.61; italics added):

‘[W]hat [Turing] needed most was a grip on rigour, on intellectual toughness, on something that was absolutely right. While the Cambridge Tripos – half ‘pure’ and half ‘applied’ – kept his in touch with science, it was to *pure mathematics* that [Turing] turned … .’

Shannon, on the other hand, as a *scientist*, never harboured any doubts about the rigour of the engineering approximations of his constructions, nor needed any confirmation on his ‘intellectual toughness’ – Turing sought the comfort of mathematically ‘rigourous’ approximations of his *scientific* constructions, whether it be the Turing Machine, Enigma machine or the Riemann ‘machine’, or whatever.
‘[W]hat [Turing] needed most was a grip on rigour, on intellectual toughness, on something that was absolutely right. While the Cambridge Tripos – half ‘pure’ and half ‘applied’ – kept his in touch with science, it was to pure mathematics that [Turing] turned … .’

Shannon, on the other hand, as a scientist, never harboured any doubts about the rigour of the engineering approximations of his constructions, nor needed any confirmation on his ‘intellectual toughness’ – Turing sought the comfort of mathematically ‘rigourous’ approximations of his scientific constructions, whether it be the Turing Machine, Enigma machine, chess machines or the Riemann ‘machine’, or whatever.

In consonant with this observation, Shannon in his constructions always assumed that playing chess with a machine was finite (but theoretically, i.e., mathematically, infinity was in the background). Relay and switching circuit constructions, on the theoretical basis of (a slightly) modified propositional algebra, lay as the underlying (complex) mechanism for chess machines (which is why Figure 5, in particular, is shown). In conclusion, I would like to add that Shannon was well aware of the differences between standard Boolean Algebra and the way propositions were derived from (even) complex circuits, the distinction and equivalences between analogue and digital computers, the truth table constructions of the calculus of propositions – hence of (Łukasiewicz) many valued logic – and the value of the Dirac delta function. He was also aware of Boole’s reliance on the exclusive interpretation of the OR connective (as distinct from Jevons’ inclusive OR) in both circuit theory and propositional calculus.

---

18 See p. 469 of Shannon’s Collected Papers (Sloane & Winer, 1993) and Novikov (1964, p. 21 is especially relevant in the Shannon finite construction sense of relay and switching circuits based chess playing machines); I am greatly indebted to the insights offered by the latter, especially on the relevance of Hilbert’s Finitism to the validity of tertium non datur and proof by contradiction. All the pioneers used freely these techniques because they worked, often implicitly, with finite systems for constructing machines. In passing, I would like to mention – especially since the Preface and Notes to Novikov (ibid) is written by Goodstein, that his valuable work on Boolean Algebra (Goodsten, 1966) was decisively important for me in appreciating Shannon’s 1938 classic.

19 Boole is mentioned on p. 474 of the classic 1938 article on Relay and Switching Circuits (Shannon, 1938; 1993), but Jevons in none of his articles; likewise, no mention of Emil Post (or Wittgenstein) even when he relies so much on the Truth Table method. Of course, there is no mention of Hailperin (1981) in 1939.
Herbert Simon

‘Chess has become a standard tool in cognitive science and artificial intelligence research (a standard “organism,” like Drosophila or Neurospora in genetics). Powerful programs … use extensive chess knowledge [but] belong to A.I., not to cognitive science.’ Simon, 1991, p. 221; italics added.

It is interesting that Simon distinguishes artificial intelligence, AI - see, above, footnote 14 - from cognitive science. For, after all, Simon was one of those who, at Dartmouth in 1956, was enthusiastically promoting AI, mainly against cybernetics (of which he was an early advocate, see (again) Velupillai, 2018, chapter 4). At the time of his death he was a Professor of Psychology and Computer Science at Carnegie Mellon University, in Pittsburgh; with much justification it is possible to say that cognitive science is (at least) a subset of Psychology and AI that of Computer Science.

So, is Simon subject to some form of schizophrenia, at least with respect to an analysis of chess games? I think, not!

Figure 6: Herbert Simon, ca. 1958
(Carnegie Mellon University & Computer History Museum; the Computer on Simon’s right is an IBM 650).
(Downloaded from the Internet/CPW article on Herbert Simon)

Simon’s most extensive analysis of chess is in the three chapters, 11, 12 & 13, of (part) 4 of the book on Human Problem Solving (Newell & Simon, 1972). In this book (and, of course,

---

20 It is impossible to talk of Herbert Simon, especially with regard to chess, without also discussing Allen Newell! Hence, this section should be read – if at all – as if it dealt with both of them (and, in part, also with Cliff Shaw, in connection with programming chess machines to play). In addition, I am convinced that Newell is part of the classic behavioural economists formed by the group of Day, March, Nelson, Simon and Winter.

21 It is the third (and last) part, of the three issues of cryptarithmetic (puzzle-solving, chapters 5,6 & 7 see also above, p.4), logic (theorem-proving, chapters 8.9 & 10) and chess. Their common framework of analysis is as Information Processing Systems (IPS) solving problems as humans would – not as
in his many articles, singly and jointly with others) Simon views chess (moves, positions and goals) as *Information Processing Systems* involved with *Human Problem Solving* (HPS) underpinning his approach to *behavioural economics* – which I refer to as *classical behavioural economics* (CBE, contrasted with *modern behavioural economics*, MBE, see Kao & Velupillai, 2013) – of which cognitive science and artificial intelligence are parts.

In §4, I view chess moves *dynamically*, in terms of the *dynamical system* approach to modelling them mathematically, in must be observed that Simon’s views chess moves dynamically as IPS, or IPS as dynamic and chess moves as a subset of this class.

Although Simon, as much as Turing and Shannon, viewed the potentialities of the machine, of artificial intelligence and of the neural network implementation of computers, he does not seem to have embraced the *Church-Turing thesis* and the concomitant *effectivity* as the only or desirable way of *algorithmizing* HPS, within CBE (see his summary of the interactions with Kleene, as the editor of *The Journal of Symbolic Logic*, and the difference with Hao Wang’s way of proving the Theorems of part I *Principia Mathematica*, in Simon, 1991, pp. 209-210). Simon has more in common with the algorithmic evolutionary economics of Nelson and Winter than with the Church-Turing thesis underpinned effective algorithms of Hao Wang.

My considered opinion is that Simon, more than Turing or Shannon, wanted to investigate why humans did what they did, the way they did it – and let machines do what they did, whatever was the way attributed to them. Thus, Turing and Shannon devised machines to implement the effective algorithms they – in effect Turing Machine implementable programs – envisaged as playing chess; Simon, on the other hand, effectivized HPS by viewing programs as heuristic search process, operating under the Gödel *completeness theorem* for propositional algebra.

‘Recent’ research results of Gandy, Sieg and others cast any amount of doubt that Turing worked with the assumption of the so-called Church-Turing theses; Shannon never assumed this thesis, especially not for the finite machines he constructed (to play chess). Thus, it is
entirely possible that Turing, Shannon and Simon worked within Hilbert’s finitary program (Novikov, ibid). I believe that in making machines to play chess, these three pioneers constructed machines, proved theorems about their functioning and achieved halting configurations of machines reaching well-defined goals (win, lose, draw). In this they were all constructivists, at least in the case of finite machines playing a finite game (such as chess).

Finally, reiterating a theme I have developed already, every mathematical economic theory must be developed for finite economies, peopled by finite agents who are, moreover, finite in number; all other aspects of production, exchange and resource constraints have respect the finiteness criteria – for example relations (in the algebraic sense). If not, we are guilty of indulging in what Ramsey (of course, in a different context) called ethically indefensible postulates.

§ 3. Elementary (Finite) Lachlan Games

“It seems that human beings love games, win or lose. It is said that in playing a game a special part of the brain operates, and I believe that games have a quality that fascinates people.”
Takeuti, 2003, p. 71; italics added.

Lachlan (1970) claimed that (p. 291; italics added):

‘[E]very theorem of T(ℛ) [of recursively enumerable (r.e.) sets] known at the present time can be proved by constructing an effective winning strategy for a suitable basic game.’

I shall use a slightly modified form of the Soare (2016), §2.5 (p. 43, ff.) definition of finite Lachlan games to interpret an alternate (computerized Chess, GO, HEX and so on) perfect information game.

The Definition of a Finite Lachlan Game

(ω = {0, 1, 2, 3, ...}; i.e., set of nonnegative integers)25

i. Player 1 (the ‘player’) constructs a finite sequence of c.e. sets {U_n}_n ∈ ω;

ii. Player 2 (the ‘opponent’) next, constructs a finite sequence {V_n}_n ∈ ω;

22 He went on to claim (p. 293):
‘[W]e do not know any theorem of T(ℛ) which is not game derivable.

23 A basic game is defined on p. 292, Lachlan (op.cit.) involving r. e. sequence of requirements and a finite combination of atomic formulas. I shall use computable enumerability (c.e.) instead of r.e. in the sequel.

24 Or chapter 16 of Soare (op.cit), in which he also states (p.221):
‘By the late 1970s, Lachlan had invented an intuitive game theory model for constructing c.e. sets which clearly revealed the intuition.’

Lachlan had, in fact, ‘invented an intuitive game theory model’ already by the late 1960s, but apart from this, I agree completely with Soare.

25 Recall that finite sets are computable.
iii. Thus, at every move of the game (of Chess) the player enumerates, at most, finitely many integers and the opponent enumerates (if possible), at most, finitely many integers; therefore, at the end of stage \( s \), a finite subset of \( \mathbb{Z} \) is the result of the actions by the player and the opponent;

iv. Each play of the game ends at the end of \( p \in \omega \) moves;

v. In advance of the game to be played, a finite c.e. sequence, \( \{R_e\}_{e \in \omega} \) of (finite) requirements (see footnote 14) are specified precisely;

vi. The opponent wins if \( Z_p \) satisfies the precisely specified requirements – the player wins, otherwise;

vii. A game situation, at the end of stage \( s \) is: \( \{U_{n,s}\}_{n < s} \land \{V_{n,s}\}_{n < s} \) and this can be effectively coded by a (+ve) integer (cf., §2.3, Soare, op.cit.);

viii. A winning strategy for a player or opponent is a function from game situation to (set of) moves, coded say as \( f \) on \( \omega \), s.t., if the player (or opponent) follows this strategy \( s(h) \) will win against any sequence of moves chosen by her/his opposite number.

Thus, given the above (substantiated) claim by Lachlan, to prove a theorem of c.e. sets, it is only necessary to construct a winning strategy in a win-lose, alternate, perfect information game. This is what is achieved in the final condition of the definition of a finite Lachlan game, now interpreted for chess\(^{26}\). Therefore, Lachlan’s Lemma 1 can be applied, with an interpretation of \( \sigma \) that is deducible from the c.e. sequence of requirements based on all of the definition elements above for finite chess games. The slightly modified Lachlan lemma 1 for finite chess games is:

**Lemma 1**

For any winning strategy of a chess game, \( \sigma \in c.e \) sets.

**Remark 1**

Recall that not all members of c.e.sets are computable. This remark is the rationale behind Corollary 1 (and Lemma 3).

**Remark 2**

Lemma 1 is particularly true of Euwe (1928; 2016).

**Conjecture 1**

Every finite game of chess is an element of (finite) constructive mathematics and the c.e. sets are strict subsets of this kind of mathematics (i.e. c.e. sets \( \subset \) finite constructive mathematical sets).

\(^{26}\) See von Neumann-Morgenstern (1953), p. 59, footnote 3 (highly relevant also for the next section) and §15.7.
Corollary 1
Chess is a Rado-type Busy Beaver Game.

Proof
Obvious, given the definitions in the Rado quote on p. 6, above and Shannon’s computation for the length of a finite chess game (see Levy, 1998a, p.6)

§ 4. Chess Moves as Dynamical Systems

‘Indeed virtually any “interesting” question about dynamical systems is – in general – undecidable’.


Simon, as discussed in the previous section, considered chess moves in terms of (dynamic) IPS in HPS senses in CBE. In this section I want to interpret chess moves as forming a dynamical system with the outcomes/goals (win, lose or draw) as members of the basins of attraction of dynamical systems; I interpret, with Turing, undecidability of the attractors of the basins of attraction of dynamical systems as the non-Halting behaviour of Turing Machine algorithms. For this I must show that the attracting set of the basins of attractors of the dynamical system interpretation of chess moves is a c.e. set.

The chess moves of player 1, say white, forms either a piecewise nonlinear dynamical system or mappings, of (in principle) large, but finite components; similarly, for player 2, who is assumed to play the black pieces on the 8x8 chess board. For convenience, I shall assume that the two players’ moves form, respectively, sets of mappings and these sets are c.e. – i.e., the elements that form the set are, each one, c.e., which means that the individual moves of the respective players are a finite sequence of c.e. sets. Therefore, the first four components – i.e., i to iv - of the definition of a finite Lachlan game are satisfied.

As for the requirements – i.e., components v & vi - of the definition of a finite Lachlan game – I assume that the basins of attraction, of the moves considered as mappings, also form c.e. sets. Then, vii & viii are trivially satisfied. Therefore:

Lemma 2
For any winning strategy of chess game, with moves considered as c.e. elements of a mapping, the basins of attraction themselves form c.e. sets.

27 Undecidable in a (computable) mathematical or (mathematical) logical sense.
**Lemma 3**

The basins of attraction are *undecidable*.

**Proof**

Based on the observation made in *Remark 1*.

**Remark 3**

In Lemma 3, the undecidability of the attractors of the basins of attraction means, first of all, that any Turing Machine algorithm may not halt because not all elements of the c.e. sets are computable; secondly, and trivially, therefore the mapping is subject to the Halting Problem of Turing Machines.

Lemmas 1 & 2 mean that the play of a chess game can be analysed in terms of computability theory or dynamical systems theory.

**§ 6. Concluding Speculations**

“I am rooting for machines. I have always been on the machines’ side.”


---

Figure 5: Little Machine constructed by *Minimax Dadamax in Person*²⁸
Peggy Guggenheim Collection, Venice, Italy
(downloaded from Wikipedia site, 7/4/2020)

---

²⁸ *Von minimax dadamax selbst konstruiertes maschinen*, 1919-1920, by Max Ernst
Machine intelligence and learnability by machines take on a new life by means of the computably enumerable sets and dynamical system interpretations of aspects of playing chess as win/lose/draw finite games. The uncritical assumption of the distinction, or the lack of clear delineation, between the (practical) finite and the (ideal) infinite has plagued machine intelligence, learnability by machines and playing chess games. It helps to clarify the role of finiteness in playing chess games, which helps in viewing the basic limits of machine intelligence and learnability by machines.

The mathematical foundations of finiteness in playing chess games is brought out clearly in re-viewing Lachlan games as steps in the direction of constructive analysis, independent of the Church-Turing thesis. This makes it possible to implement algorithms in studying machine intelligence and learnability by machines without assuming the Church-Turing thesis – and the dimensional difficulties are highlighted by bringing in the (finite) problems of the Busy Beaver Games.

I prefer to restart the program of studying machine intelligence in terms of the suggestions made by House & Rado (1964), in the context of the noncomputability of finite sets, as I have done for finite chess games and the resulting c.e. sets and the basins of attraction of dynamical systems modelling (practically) the dynamics of chess moves and the outcome space.

I also prefer to concentrate on the learnability of finite machines proving propositions without assuming axioms. I have not assumed any axioms in §3 & §4, although they seem to be part of the repertoire of the mathematician using c.e. sets and dynamical systems. All one needs are rules – not necessarily axioms. This is illustrated in the mathematical appendix.

When Simon, et. al., (Levy, 1998b, p. 91) stated the following:

‘Chess is the intellectual game par excellence. …. If one could device a successful chess machine, one would seem to have penetrated to the core of human intellectual endeavour.’

they seem to have understood that machine intelligence and learnability by machines depended on penetrating ‘the core of human intellectual endeavour.’ The main thrust of this

---

29 Recall that machine intelligence is the original phrase for artificial intelligence; I prefer the ‘old fashioned’ phrase, especially because it was not introduced to differentiate the concept from cybernetics.
paper, and the mathematical appendix is based on humans penetrating the core of the machine's intellectual endeavour. Unless research and realizations of machine intelligence and learnability by machines comes to terms with the machine’s intellectual endeavour, then like the dependence of algorithmic formulation assuming the Church-Turing thesis, humans will be chasing a will-o’-the-wisp!

§ A. Appendix: Mathematical Notes

![Image of a triangle](image)

**Theorem:**
Every acute-angled triangle can be made into, or from, three isosceles triangles\(^{30}\).

**Proof:**
See Figure 9.

The above theorem is ‘proved’ visually – without words or symbols. Now, take an example of proof of an Euclidean proposition by a machine, given only the rules of Euclidean geometry, using a program – i.e., algorithm - devised by Marvin Minsky\(^{31}\). The ‘rules of Euclidean geometry’ include the ‘fact’ that the three angles of a triangle must sum to 180°; this is equivalent to the axiom of parallels – the removal of which leads to non-Euclidean geometries (of Bolyai, Lobachevsky, Gauss, etc).

Axioms, like the ‘character of an opponent’, are human concepts – like the axiom of choice, the axiom of completeness, etc. Humans develop axiom systems, seek formalizations, and the like, to solve problems or generate theorems. This is, I think, what Turing (1992a; 1947, p. 104; italics added) meant, when he stated:

‘[I]f a mathematician is confronted with such a problem [i.e., that with certain logical systems, there is no test that can be applied which will divide propositions, with certainty,}

---

\(^{30}\) Adapted from MacHale (2008).

\(^{31}\) Michie, 1986, p.11 & p.23; the Minsky role is emphasized in McCorduck, 2004, p. 126 and also in note 21, p. 352 of MacKenzie, 2001 (except that there is an obvious typographical error in referring to p.106 instead of 126; it may be that MacKenzie is referring to an earlier edition of McCorduck).
into classes that were provable and unprovable] he would search around and find new methods of proof, so that he ought eventually to be able to reach a decision about any given formula.”

This, coupled with the idea in Turing (1992b; 1947, p. 127; italics added), that:
“[I]ntellectual activity consists mainly of various kinds of search.’

What do machines do? Do they search over a space autonomously – i.e., without human intervention? Do they formulate axioms to facilitate the proving of theorems? Could they have developed the axiom of completeness due to their dissatisfaction with a non-constructive mathematics dependent on the axiom of choice? Can they develop an intuitionistic constructive mathematics by themselves, i.e., without the intervention of a Brouwer or a Martin-Löf.

I like to think that this is the defining characteristic between humans and machines – but I am not sure. After all, there are mathematical and physical systems which have defied axiomatisation – Bishop’s constructive mathematics, Feynman diagrams, even Brouwer’s intuitionistic constructivism (despite Heyting’s ‘compromises’), are examples of the failure of the axiomatizing crusade – which has nothing to do with machine intelligence or learnability by machines.

References


De Rosa, Alessandro (2019, edited), Ennio Morricone: In His Own Words, Oxford University Press, Oxford.

32 In his/her mind!


