Undecidability in the Imitation Game

YUZURU SATO[1,*] and TAKASHI IKEGAMI[2]
[1]*Brain Science Institute, Institute of Physical and Chemical Research (RIKEN), 2-1 Hirosawa, Wako, Saitama 351, Japan*
[2]*Graduate School of Arts and Science, University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153, Japan*

Abstract. This paper considers undecidability in the imitation game, the so-called Turing Test. In the Turing Test, a human, a machine, and an interrogator are the players of the game. In our model of the Turing Test, the machine and the interrogator are formalized as Turing machines, allowing us to derive several impossibility results concerning the capabilities of the interrogator. The key issue is that the validity of the Turing test is not attributed to the capability of human or machine, but rather to the capability of the interrogator. In particular, it is shown that no Turing machine can be a perfect interrogator. We also discuss meta-imitation game and imitation game with analog interfaces where both the imitator and the interrogator are mimicked by continuous dynamical systems.

Key words: analog computation, dynamical systems, imitation game, Turing machine, undecidability

1. Introduction

The 'man–machine problem' is raised in the imitation game proposed by Turing (1950), the so-called Turing Test. The imitation game is a party game to determine from a teletype-based communication whether an out-of-sight player is a man or a woman. Figure 1 shows a schematic view of the Turing Test. A man (A), a woman (B), and an interrogator (C) are the players of the imitation game. The object of the game for (C) is to identify the opponent player (X) as (A) or (B). (A)'s object in the game is to cause (C) to make the wrong identification by telling clever lies in the conversation while (B) helps (C) to identify her correctly.

In order to consider the man–machine problem, the man (A) is replaced by a machine in the game. In this case, what would happen? Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? This question can replace 'can machines think?' Although humans are different from machines in a trivial sense, Turing challenged this opinion as it related the machine thought problem. By performing a thought experiment that had inductive evidence of 'think', he presented a kind of paradox in the cognitive sciences. The framework proposed by Turing became a philosophical basis for the classical studies on artificial intelligence (Weizenbaum, 1966; Hofstadter and Denett, 1981).

A large number of studies related to the Turing Test[1] consider the capability of imitated player (human), the imitating player (machine), and the differences

*Corresponding author

Minds and Machines **14:** 133–143, 2004.

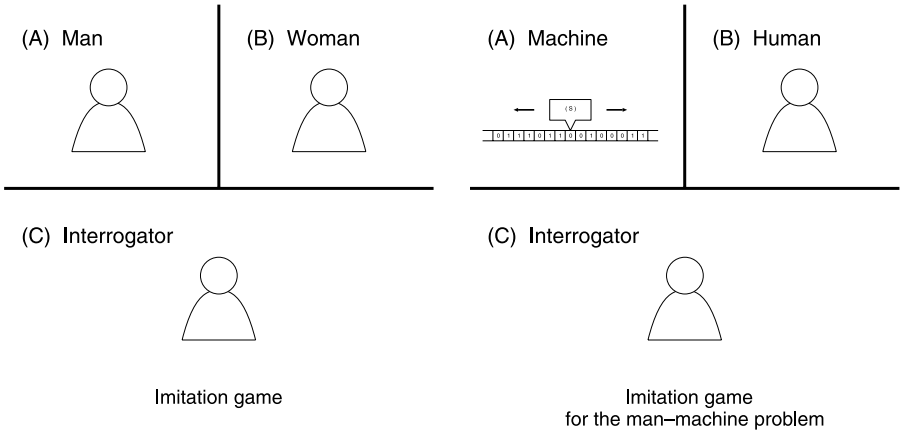

Figure 1. The imitation game: A man (A), a woman (B), and an interrogator (C) are the players of the game. The man is replaced by a machine in the right figure.

between the two (see, e.g., Searle, 1980). However it is Turing's argument that one cannot distinguish humans from machines in a situation like the imitation game. Thus, the Turing Test can be interpreted as a study of the subjectivity of the interrogator who participates in the game as a player. Here, we propose that the validity of the Turing Test is not attributed to the capability of the imitated player (human) or the imitating player (machine) but rather to the capability of the interrogator. In our model of the Turing Test, the imitator (A) and the interrogator (C) are formalized as Turing machines (Turing, 1936). We analyze the pattern of conversation between the imitator (A) and the interrogator (C) and suggest that the role of the interrogator is much more important than previously supposed. This viewpoint for the imitation game was pointed out in Premack and Woodruff (1978) with the theory of mind and in Watt (1996) with the inverted Turing test.

In this paper, we introduce a computational theoretical approach to the imitation game and give a simple formalization of it in Section 2. In Section 3, we give a concrete example of the imitation game on networks and apply the results demonstrated in Section 2. In Sections 4 and 5, based on the results in Section 2, we consider meta-imitation game and imitation game with analog interfaces where both the imitator and the interrogator are mimicked by continuous dynamical systems.

2. Undecidability in the Imitation Game

2.1. CAN MACHINES THINK 'CAN MACHINES THINK?'

We consider the undecidability of the imitation game and answer 'no' to the above question. The imitator (A) and the interrogator (C) are modeled as Turing machines[2] (see Figure 2).

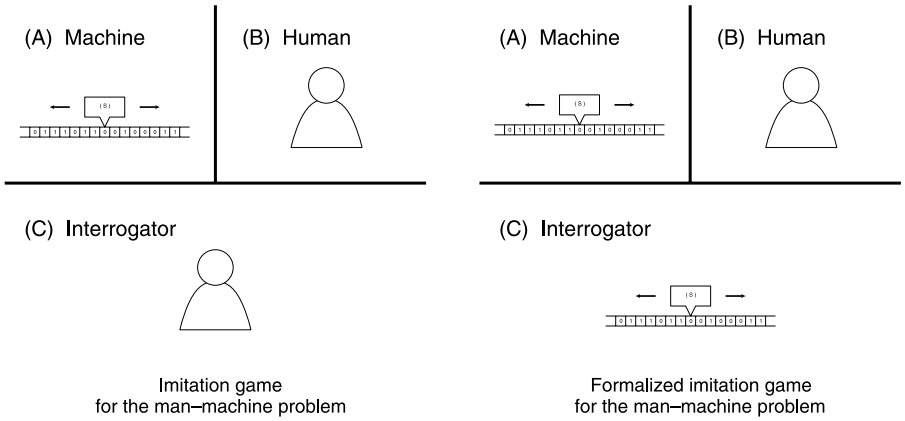

Figure 2. Formalized imitation game: the interrogator is replaced by a Turing machine.

In order to analyze the computational capability of the interrogator, we model the imitation game with the following conditions:

1. The imitator (A) is a Turing machine.
2. The interrogator (C) is a Turing machine.
3. The player (X) is either an imitator (A) or a human (B).
4. The interrogator (C) is given a complete blueprint M of the imitator (A).
5. The human (B) is not given the blueprint of the imitator (A). Thus, he or she cannot emulate the imitator (A) intentionally.
6. The interrogator (C) halts and outputs `Yes` when it finds that the player (X) is the target Turing machine M, otherwise it halts and outputs `No`.
7. When the imitator (A) does not halt, the interrogator (C) does not have to halt.

The question for the interrogator in the original imitation game

‘Is player (X) a machine or not?’

replace an easy one

‘Is player (X) the target Turing machine M or not?’

and a Turing machine answers `Yes` or `No` for this question by using the information from the teletype-based communication.

The above conditions 1–7 give the interrogator a greater advantage over the imitator than in the original setting. Nevertheless, we reach undecidability for the interrogator’s decision of human or machine.

2.2. DIAGONALIZATION: A MACHINE FOR MACHINE DETECTION

Assume that the imitator tries to answer $a_n \in \Sigma^*$ against the interrogator’s nth question $q_n \in \Sigma^*$. Σ is the set of alphabets on a ‘teletype’, and Σ^* is the set of all

Table I. Flow of communication in the imitation game

q_1		q_2		q_3		q_4		
$\downarrow$	$\nearrow$	$\downarrow$	$\nearrow$	$\downarrow$	$\nearrow$	$\downarrow$	$\nearrow$	$\cdots$
a_1		a_2		a_3		a_4		

strings on that alphabet. The conversation between the imitator and the interrogator is characterized by the flow of the sequences a_n's and q_n's as shown in Table I.

We can add a delimiting character '$\sqcup$' to the alphabet Σ without loss of generality. Then we define conversation sequences $C = \cup_{n=1}^{\infty} C_n \subset (\Sigma + \{\sqcup\})^*$, where

$$C_n = \begin{cases} \Sigma^* & (n = 1), \\ \Sigma^*(\sqcup \Sigma^* \sqcup \Sigma^*)^{n-1} & (n > 1), \end{cases}$$
$$c_n = q_1 \sqcup a_1 \sqcup q_2 \sqcup a_2 \cdots \sqcup a_{n-1} \sqcup q_n \in C_n.$$

Here $c_n \in C_n$ is a sequence in which q_n's and a_n's are concatenated with delimiting character $\sqcup$'s.

Next, we design the decision problem TEST.

DEFINITION 2.1 (TEST). We define a language L_M as follows:

$$L_M = \{c' \in \Sigma^*(\sqcup \Sigma^* \sqcup \Sigma^*)^{n-1} | n \geq 1, \text{ given a Turing machine } M, \text{ if } M \text{ halts for input } c_n \in C_n, \text{ then } c' = c_n \sqcup M(c_n)\}.$$

The answer $a_n = M(c_n) \in \Sigma^*$ is generated so that a Turing machine M acting as an imitator tries to deceive the interrogator on the nth round. If the conversation sequence between an imitator and an interrogator belongs to the set L_M, the interrogator will decide that the imitator is the target machine M. The decision problem TEST associated with this imitation game is stated as, 'given M and $x \in \Sigma^*(\sqcup \Sigma^* \sqcup \Sigma^*)^i$, $x \in L_M$?'.

THEOREM 2.2. *TEST is undecidable.*

Proof. See Appendix A. □

Since TEST contains the halting problem of Turing machines, it is undecidable. In short, if a perfect interrogating algorithm exists, we can make an algorithm to outwit it by using itself and this leads to a contradiction[3]. This implies that there is no effective procedure to decide whether the player (X) is the imitator (A) or not, even when using the conditions 1–7, which gives the interrogator a greater advantage over the imitator than in the original setting.

3. An Example: Network Games

'Quake' is a multi-player network game of gun fighting. The server machines for this game are on the Internet, and many players connect to them to play the game.

Recently, the administrator of the game servers have become worried about a malignant outbreak of cheaters. The cheaters use cheat-programs which play the game better than average human players. Logging on to the game, the cheat-programs beat all the human players by using their overwhelming abilities and always win the game. After discovering this, the administrator created a check program to use against these cheat-programs and tried to prevent them from joining the game. However, this yielded a new problem. Excellent human players who have higher ability are judged to be cheat-programs and blocked by the check program. Consequently, the skilled human players cannot play the game on any 'secured' servers.

This phenomenon is very similar to our model of the imitation game. We can say that there is no check program which can distinguish a skilled human player from a cheat program even if we have a complete blueprint of the cheat program.[4]

4. Meta-Imitation Game

Turing posed the question, 'can machines think?' and 'can we tell the difference between a human and a machine in the imitation game?' (Turing, 1950). In Section 2, we have shown that no Turing machine can be a perfect interrogator, even if it is given a complete blueprint of the imitator. However, whether a human can play the role of the interrogator successfully or not is still an interesting question: 'Can an interrogator successfully distinguish humans from Turing machines?' We can

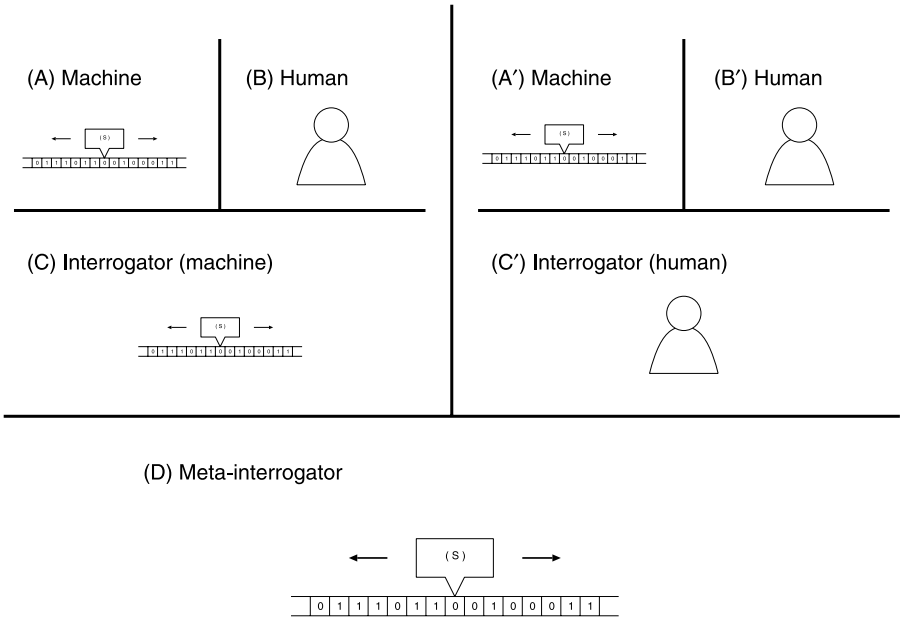

Figure 3. Meta-imitation game: a meta-interrogator observes plays of imitation games.

say that if an interrogator can always distinguish humans from Turing machines, the interrogator cannot be a Turing machine.

Furthermore, we might be able to use this as a means of distinguishing humans from machines, that is, whether they execute the imitation game successfully or not determine their identity. This could be formalized as a 'meta-imitation game' to test the interrogator (see Figure 3). However, we can expect that the meta-imitation game is undecidable for the meta-interrogator anyway, since a Turing machine cannot by any means be the perfect interrogator of the imitation game.[5]

5. Imitation Game with Analog Interfaces

In the original imitation game, non-symbolic interactions between the imitator and the interrogator are neglected.[6] However, if we include analog signals in the imitation game, the problem of the imitation game might become more difficult. For example, a question such as 'what does this smell like?' will now be allowed.[7] The interrogator now has to distinguish a human from a machine by images, sounds, embodied gestures or 'randomness'[8] without being constrained to use only a teletype.[9]

Here, we focus on the computability of this 'analog imitation game'. Following Turing, we consider the man–machine problem by using continuous dynamical systems. We could pose the question 'can dynamical systems think?' using Turing's framework to consider 'think' and 'dynamical systems', that is well-defined mathematical notion. Since there are many studies that try to recapture 'computation' based on continuous dynamical systems (Crutchfield and Young, 1989; Pollack, 1991; Moore, 1998; Sato et al., 2001), we can describe the new form of the problem using dynamical systems with computational abilities.

In Moore (1990), it is shown that there exists two-dimensional mappings and three-dimensional flows that contain an universal Turing machine topologically embedded in the symbolic dynamics. Any non-trivial questions about these dynamical systems, such as existence of sensitive dependence, measures of basins of attraction, or dimensions of attractors, are all undecidable with Rice's theorem (1953). Even in a spatio-temporally continuous system such as differential equations, analog circuits, or neural networks,[10] we can similarly find undecidability in terms of analog computation (Pour-El and Richards, 1989; Blum et al., 1998; Seigelmann, 1999; Campagnolo et al., 2000).

These observations lead us to expect that no dynamical system can be a perfect interrogator in the analog imitation game even if it is given a complete blueprint of the imitator[11] in the similar way to in Section 2. Although the interrogator is an analog machine which can directly 'operate' real numbers and evolve in continuous-time, the imitator can also be analog and, again, can cheat the interrogator.

It is a common approach to model human cognition by using dynamical systems (see, e.g., Crutchfield, 1998; Gelder, 1998); however, dynamical systems cannot execute the analog imitation game successfully. Thus, whether a human can play

the role of the interrogator successfully or not in this analog imitation game leaves an interesting question: 'Can a human distinguish thinking dynamical systems from a non-thinking ones?' We can say that if an interrogator can always distinguish humans from dynamical systems in the analog imitation game, the interrogator cannot be a dynamical system.[12]

6. Summary

The proposal presented here is that when two systems that have exactly the same computational ability play an imitation game with human, it is impossible for the interrogator (C) to distinguish the imitator (A) from human (B). In other words, let s be a system that can work as computing machinery and let S be the set of the computing machineries including systems s. Then we can say that no system $s' \in S$ can be a perfect interrogator that can decide whether the opponent player (X) is the given $s \in S$ or not and if the interrogator can distinguish humans from systems S for all s, the interrogator cannot belong to S. Thus, Turing's claim that humans cannot distinguish men from women in the imitation game is not falsified.

Appendix A. Proof for the Theorem

Proof. For the sake of contradiction, we make the assumption that TEST is decidable at the nth round, that we have three-tuples of (M, c_n, a_n), and that the player returns an answer at the nth round. Now, there exists a Turing machine which plays the role of the interrogator in the imitation game. It can decide whether the opponent player (X) is the target machine M or not and always halts within a finite steps. Such a Turing machine contains a subroutine algorithm called IS-MACHINE:

$$\texttt{IS-MACHINE}\ (M, x, y) = \begin{cases} \texttt{Yes} & (x_{\sqcup} y \in L_M), \\ \texttt{No} & (\text{otherwise}). \end{cases}$$

IS-MACHINE can correctly solve a membership problem for L_M on arbitrary M, x, y and always halts within a finite number of steps. If machine M halts for input x and outputs y, IS-MACHINE decides that the player (X) is the machine M and outputs Yes. If M halts for input x and outputs $z(\neq y)$, IS-MACHINE decides that the player (X) is not the machine M and outputs No. Otherwise, if M does not halt for input x, IS-MACHINE decides that the player (X) is not the machine M and outputs No (because the opponent actually returns the answer y).

Given such an interrogator that contains IS-MACHINE, we can construct an algorithm D as an imitator in the imitation game that has the following behavior:

$$\texttt{D}(x) = \begin{cases} \text{goto LOOP } (\texttt{IS-MACHINE}\ (x, x, \texttt{I'm not the machine!}) = \texttt{Yes}), \\ \text{write 'I'm not the machine!'} \quad (\text{otherwise}). \end{cases}$$

If IS-MACHINE $(x, x,$ I'm not the machine!$) =$ Yes, D falls into an infinite loop, otherwise D answers 'I'm not the machine!' to the interrogator. We can now

get a contradiction by examining the behavior of D(D) in the case where we give a conversation sequence D for the machine D.

(i) If D(D) halts and outputs 'I'm not the machine!', then
IS-MACHINE (D, D, I′m not the machine!) = Yes,
D(D) does not halt. That is a contradiction.
(ii) If D(D) does not halt, then
IS-MACHINE (D, D, I′m not the machine!) = No,
D(D) halts and outputs 'I'm not the machine!'. That is a contradiction.

Consequently IS-MACHINE cannot be a Turing machine that is correct and always halts. Thus, TEST is undecidable. □

Note that the standard diagonalization technique is used here (see Anderlini, 1990; Dowling, 1990 for the similar applications). Since D(D) is different at every

Table II. The diagram of M_i vs. c_n^j: we make lexically ordered numbering of all Turing machines and all $c_n \in C_n$, and denote ith Turing machine M_i, and jth c_n c_n^j[a]

	c_n^1	c_n^2	c_n^3	$\cdots$	c_n^j	$\cdots$
M_1	a_n^{11}	a_n^{12}	a_n^{13}	$\cdots$	a_n^{1j}	$\cdots$
M_2	a_n^{21}	a_n^{22}	a_n^{23}	$\cdots$	a_n^{2j}	$\cdots$
M_3	a_n^{31}	a_n^{32}	a_n^{33}	$\cdots$	a_n^{3j}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
M_i	a_n^{i1}	a_n^{i2}	a_n^{i3}	$\cdots$	a_n^{ij}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

[a] The outputs $M_i(c_n^j)$ are denoted as a_n^{ij}.

Table III. Diagonalization: let the lexical order number of D be k, then $\mathtt{D} = M_k$, $\mathtt{D} = c_n^k$[a]

	c_n^1	c_n^2	c_n^3	$\cdots$	$\mathtt{D} = c_n^k$	$\cdots$
M_1	$\bar{a}_n^{11}$					
M_2		$\bar{a}_n^{22}$				
M_3			$\bar{a}_n^{33}$			
$\vdots$				$\ddots$		
$\mathtt{D} = M_k$					$\mathtt{D}(\mathtt{D}) = \bar{a}_n^{kk}$	
$\vdots$						$\ddots$

[a] The output D(D), denoted as $\bar{a}_n^{kk}$, is different at every position from the diagonal of the matrix M_i versus c_n^j for all k.

position from the diagonal of the matrix M_i versus c_n^j (see Tables II and III), D is different from all M_i[13]. In short, if an effective procedure IS-MACHINE exists, we can make an algorithm such as D to outwit IS-MACHINE by using IS-MACHINE itself.

Acknowledgements

The authors would like to thank Professor J.P. Crutchfield (Santa Fe Institute) for his useful comments. The author (YS) would also like to thank the colleagues at Santa Fe Institute. This work was supported by the Special Postdoctoral Researchers Program at RIKEN.

Notes

[1] See French (2000) for a survey.

[2] See Bradford and Wollowski (1993) for a computational complexity analysis. Since the origin of studies on the interactive proof system is the Turing test, the formalization in Bradford and Wollowski (1993) is quite natural.

[3] For similar applications of diagonalization, see Anderlini (1990) on rationality of game players and Dowling (1990) on computer viruses and vaccines.

[4] Given this example, one becomes interested in the imitation game on networks. In this network imitation game, many interrogators will try to identify each other and an interrogator will also be an imitating/imitated player. They can observe plays of other imitation games and use the information from conversations among other two players: A player (X) has a conversation with a player (Y) and decides 'the player (Y) is a machine', while a player (Z) observing this conversation decides that 'the player (X) is a machine and the player (Y) is a human', and all the chain of these decisions and conversations can be the information for the fourth player (W). Similar frameworks are proposed in Forner (1997) and Mauldin (1994). These observations lead us to investigate the meta-imitation game. Another direction is to focus on the continuity of the interface in the example of network games. This leads us to investigate the analog imitation game. We discuss the computability of meta-imitation game and the analog imitation game in the following sections.

[5] As a computational theoretical approach, we can give an oracle for the interrogator to solve TEST and introduce a hierarchy of the interrogators corresponding to the hierarchy of computability.

[6] The interface of an early model of the imitation game was implemented as chess-playing (Turing, 1948), but this would be same as the teletype-based communication. It is possible to discuss that the information from teletypes or written letters are analog signal for human players, such as light towards player's retina, to be sure, but here, we discuss the structures of the interfaces in the game.

[7] This type of communication existed in ancient Japan where non-verbal communication was attempted through the exchange of perfumed letters. See, for example, 'The Tale of Genji' by Murasaki Shikibu.

[8] Human random generation has been studied in psychology (Wegenaar, 1972) and it is known that humans cannot generate perfectly random sequences. As for human's wrong intuitions for prior probability, see Tversky and Kahneman (1974).

[9] In this general setting, the symbol grounding problem (Harnad, 1990, 2001) essentially arises for both the imitator and the interrogator.

[10] Turing commented on the possibility of using an analog circuit for the imitator and concluded that when the interface is discrete, the difference between digital machine and analog machine would be indistinguishable (Turing, 1950). Our discussion is for the case where both machines and interfaces are continuous.

[11] We would need an additional assumption that the Kolmogorov complexity of the blueprint of the imitator should be finite.

[12] We can construct a meta-game for this analog imitation game similarly to in Section 3.

[13] The proof in Sato and Ikegami (1999) is partly revised here.

References

Anderlini, L. (1990), 'Some Notes on Church's Thesis and the Theory of Games', *Theory and Decision* 29, pp. 19–52.

Blum, L., Cucker, F., Shub, M. and Smale, S. (1998), *Complexity and Real Computation*, New York: Springer-Verlag.

Bradford, P.G. and Wollowski, M. (1993), 'A Formalization of the Turing Test', *Proceedings of 5th Midwest Artificial Intelligence and Cognitive Science Conference*, pp. 83–87.

Campagnolo, M.L., Moore, C. and Costa, J.F. (2000), 'Iteration, Inequalities, and Differentiability in Analog Computers', *Journal of Complexity* 16(4), pp. 642–660.

Crutchfield, J.P. (1998), 'Dynamical Embodiment of Computation in Cognitive Processes', *Behavioral and Brain Sciences* 21(5), pp. 635–637.

Crutchfield, J.P. and Young, K. (1989), 'Inferring Statistical Complexity', *Physical Review Letters* 63, pp. 105–108.

Dowling, W.F. (1990), 'Computer Viruses: Diagonalization and Fixed Points', *Notice of the AMS* 37, pp. 858–861.

Forner, L. (1997), 'Entertaining Agents: A Sociological Case Study', *Proceedings of International Conference on Autonomous Agents 97*, pp. 122–129.

French, R.M. (2000), 'The Turing Test: The First Fifty Years', *Trends in Cognitive Sciences* 4(3), pp. 115–121.

Harnad, S. (1990), 'The Symbol Grounding Problem', *Physica D* 42, pp. 335–346.

Harnad, S. (2001), 'Minds, Machines, and Turing: The Indistinguishability of Indistinguishables', *Journal of Logic, Language, and Information* 9(4), pp. 425–445.

Hofstadter, D.R. and Dennett, D.C. (1981), *The Mind's I*, Basic Books.

Mauldin, M. (1994), 'Chatterbots, Tiny MUDs, and the Turing Test', *Proceedings of National Conference of Artificial Intelligence 94*, pp. 16–21.

Moore, C. (1990), 'Unpredictability and Undecidability in Dynamical Systems', *Physical Review Letters* 64, pp. 2354–2357.

Moore, C. (1998), 'Dynamical Recognizers: Real-Time Language Recognition by Analog Computers', *Theoretical Computer Science* 201, pp. 99–136.

Pollack, J.B. (1991), 'The Induction of Dynamical Recognizers', *Machine Learning* 7, pp. 227–252.

Pour-El, M.B. and Richards, J.I. (1989), '*Computability in Analysis and Physics*', Berlin: Springer-Verlag.

Premack, D. and Woodruff, G. (1978), 'Does the Chimpanzee Have a Theory of Mind?', *Behavioral and Brain Sciences* 1, pp. 516–526.

Rice, H.G. (1953), 'Classes of Recursively Enumerable Sets and Their Decision Problem', *AMS* 89, pp. 25–59.

Sato, Y. and Ikegami, T. (1999), 'Undecidability of the Imitation Game', *Proceedings of Symposium on Imitation in Animals and Artifacts*, pp. 157–159.

Sato, Y., Taiji, M. and Ikegami, T. (2001), 'Computation with Switching Map Systems: Nonlinearity and Computational Complexity', *Santa Fe Institute working paper*, WP01-12-083.

Searle, J.R. (1980), 'Minds, Brains, and Programs', *Behavioral and Brain Sciences* 3, pp. 417–424.

Siegelmann, H.T. (1999), *Neural Networks and Analog Computation*, Boston: Birkhauser.

Turing, A.M. (1936), 'On Computable Numbers, with an Application to the Enentscheidungsproblem', *Proceedings of the London Mathematical Society* 2(42), pp. 230–265.

Turing, A.M. (1948), 'Intelligent Machinery', in B. Meltzer and D. Michie, eds., *National Physical Laboratory Report*, *Machine Intelligence* 5, pp. 3–23, 1969.

Turing, A.M. (1950), 'Computing Machinery and Intelligence', *Mind* 59(236), pp. 433–460.

Tversky, A. and Kahneman, D. (1974), 'Judgement under Uncertainty: Heuristics and Biases', *Science* 185, pp. 1124–1131.

van Gelder, T. (1998), 'The Dynamical Hypothesis in Cognitive Science', *Behavioral and Brain Science* 21, pp. 1–14.

Wagenaar, W.A. (1972), 'Generation of Random Sequences by Human Subjects: A Critical Survey of Literature', *Psychological Bulletin* 77, pp. 65–72.

Watt, S.N.K. (1996), 'Naive Psychology and the Inverse Turing Test', *Psycholoquy* 14(7), p. 1.

Weizenbaum, J. (1966), 'ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine', *Communication of the ACM* 9, pp. 36–45.