International Journal of Innovative Computing 16(1) 43–47

UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

The Halting Problem in Complexity Theory: A Review

Muhammad Iqbal Habibie[1*] & Yusliza Yusoff[2]
Faculty of Computing
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
Email: habibie@graduate.utm.my[1]; yusliza@utm.my[2]

Submitted: 17/6/2025. Revised edition: 14/12/2025. Accepted: 5/1/2026. Published online: 10/6/2026
DOI: https://doi.org/10.11113/ijic.v16n1.560

***Abstract*—The Halting Problem is a pivotal concept in computational theory, questioning whether a general algorithm can determine if a program halts or runs indefinitely on a given input. While foundational works have established its undecidability, a critical comparative evaluation of how the problem has evolved across theoretical, educational, quantum, and dynamical domains remains lacking in the literature. This gap limits comprehensive understanding and cross-domain applicability for researchers. Therefore, this paper aims to address that void by analyzing and synthesizing five influential contributions: Turing's 1937 foundational proof, Pavlotskaya's 2002 mathematical formalization, Sipser's 2013 pedagogical perspective, Aaronson's 2016 exploration into quantum undecidability, and Cotler and Rezchikov's 2024 application in computational dynamical systems. Through structured analysis of their methodologies, assumptions, and implications, this paper highlights the strengths, limitations, and evolving interpretations of the Halting Problem. Results confirm that its undecidability has far-reaching effects on algorithm design, software verification, and emerging areas like quantum computing and chaotic systems. By identifying key gaps such as lack of empirical validation and practical tools, this paper proposes future directions including approximation algorithms, interdisciplinary modeling, and AI safety applications. This study thus offers a consolidated foundation for researchers seeking to explore the Halting Problem's role across both classical theory and modern computational frontiers.**

***Keywords*—Halting Problem, Complexity Theory, Undecidability, Turing Machine, Computability**

I. INTRODUCTION

The Halting Problem, first formalized by Turing (1937), poses a fundamental question in computational theory where it asks if an algorithm can determine whether a given program *P* halts or runs forever depending on input *I*. Turing's proof of its undecidability revealed inherent limits of computation, profoundly influencing complexity theory, algorithm design, software verification, and more. Complexity theory, a core area of computer science, studies the time, space, and power needed for computation, classifying problems as decidable, undecidable, or tractable based on solvability and efficiency.

This research paper explores the halting problem's significance within complexity theory through a critical analysis of five known works: Turing's 1937 paper on computable numbers, Pavlotskaya's 2002 study on Turing machines and undecidability, Sipser's 2013 textbook chapter on undecidability, Aaronson's 2016 exploration of quantum undecidability, and Cotler and Rezchikov's 2024 study on computational dynamical systems.

The halting problem remains an important topic as it helps the process of creating modern automated tools, theoretical models, and tackling emerging domains like quantum computing and dynamical systems. This paper is organized as follows: the background traces the problem's history and context; the methodology outlines our review approach; results and analysis synthesize findings; and the discussion proposes future directions.

The objective of this paper is to identify the contributions, strengths, and limitations of five pivotal works on the Halting Problem and its role in complexity theory. This critical synthesis aims to guide future research and improve the theoretical and practical relevance of undecidability in computing domains.

II. BACKGROUND

Complexity theory studies the resources required for solving computational problems, typically measured in time, space, and computational power. Problems are classified as decidable (solvable with guaranteed termination) or undecidable (no general algorithm exists for all inputs). The Halting Problem belongs to the latter. Problems are also classified further by their tractability. Tractable problems are solvable in polynomial time ($O(n^k)$), while intractable problems are solvable in exponential time ($O(2^n)$), making them impractical for large inputs. Therefore, the importance of complexity theory is seen applied in modern algorithm design, system optimization, and theoretical limits of computation.

A. Introduction of the Halting Problem

The halting problem was introduced by Alan Turing (1937) where it asks the question "given program *P* and input *I*, does program *P* halt/terminate or run indefinitely on input *I*?" To address this, Turing developed the Turing machine, an abstract model of computation comprising of an infinite tape divided into cells, a read/write head to manipulate symbols, a finite set of states, and a rule table governing its transitions. Turing's paper, "On Computable Numbers, with an Application to the Entscheidungsproblem," emphasizes a diagonalization argument to prove the problem's undecidability, where no general algorithm can determine halting for all possible programs and inputs.

In 1900, David Hilbert posed the Entscheidungsproblem, asking for an algorithm to decide the truth of mathematical statements. Turing's (1937) work addressed this problem and proved the halting problem's undecidability, which by extension, also establishes the undecidability of Hilbert's question. These established computability limits show that some problems are beyond algorithmic solutions.

B. The Halting Problem's Impact in the Modern Day

In the modern day, Pavlotskaya (2002) built on this by using rigorous mathematical proofs to explore the connection between Turing machines and undecidability, through a theoretical framework, thus emphasizing formal logic of the halting problem. Sipser (2013) published a textbook titled "Introduction to the Theory of Computation". His work provided a pedagogical lens, detailing undecidability in a chapter of the book. He used examples and reductions to clarify the halting problem's implications, thus bridging theory and education.

Aaronson (2016) then extended the discussion to quantum computing in "The Ghost in the Quantum Turing Machine." Their work introduces quantum Turing machines, extending the traditional models by incorporating quantum states, superposition, and entanglement, further exploring undecidability in this context. Recently, Cotler and Rezchikov (2024) connected the halting problem to computational dynamical systems, examining termination in iterative, often chaotic, processes relevant to physics and complex systems. These works trace the problem's journey from foundational theory to modern, interdisciplinary applications.

The halting problem's undecidability has shaped multiple domains. It defines the boundaries of what algorithms can achieve in terms of their computability, and it informs the problem classification by distinguishing the decidability and undecidability of the problem. The halting problem also helps in limiting automated tools for program verification and debugging. Meanwhile, emerging fields like quantum computing and dynamical systems explore the relevance of the halting problem to new models like quantum algorithms and chaotic iterations.

This further underline the ongoing influence of the problem's centrality on theoretical and applied computer science, enabling us to trace its evolution, which can be seen in Table I.

TABLE I. KEY CONTRIBUTIONS IN HALTING PROBLEM RESEARCH

Year	Author(s)	Key Contribution
1937	Turing	Defined Turing Machines that proved halting problem undecidability
2002	Pavlotskaya	Linked Turing machines to undecidability via mathematical proofs
2013	Sipser	Clarified undecidability in a pedagogical framework
2016	Aaronson	Explored halting problem in quantum Turing machines
2024	Cotler & Rezchikov	Applied halting problem to computational dynamical systems

III. METHODOLOGY

This study employs a critical literature review to analyze the five pivotal papers on the halting problem and assess their contributions to complexity theory. Each work is analyzed according to three criteria:

- Contribution to understanding undecidability in complexity theory.
- Methodological rigor and clarity.
- Identifiable strengths and limitations.

This paper examines each work's literature review for scope and context, methodology for rigor and approach, and conclusions for clarity and impact. The review mainly focuses on their depth, precision, and relevance to the halting problem and its implications.

A. Turing's Work

Turing's (1937) paper, published in the Proceedings of the London Mathematical Society, introduced the Turing machine and proved the halting problem's undecidability.

In the Literature Review of Turing's work, the scope and framing are assessed by considering the emerging field of computability and references to prior mathematical challenges, such as Hilbert's work. Next, Turing's Methodology is evaluated by examining the rigor of the definition of computation through Turing machines—encompassing states, tape, and transition rules—and the logical structure of the diagonalization argument, which demonstrates a contradiction. Finally, in Turing's Conclusions, the clarity of the undecidability proof and its foundational impact on theoretical computer science are analyzed.

B. Pavlotskaya's Work

Pavlotskaya's (2002) work, published in Mathematical Notes, explored the halting problem through formal mathematical proofs, connecting Turing machines to undecidability.

In the Literature Review of Pavlotskaya's work, the breadth and quality of citations, including Turing and related theoretical works, are evaluated to establish the work's context. Then, the precision of the authors' axioms for Turing machines, the step-by-step validity of the authors' derivations, and the mathematical rigor of the authors' approach, which is stated in the Methodology, are investigated. Finally, in the Conclusions, the coherence of the authors' findings, the reinforcement of undecidability, and the contribution to complexity theory are reviewed.

C. Sipser's Work

Sipser's (2013) chapter from Introduction to the Theory of Computation provided a pedagogical perspective on undecidability.

In the Literature Review of Sipser's work, the integration of prior works, including Turing and others, is assessed to frame undecidability for learners. Then, the clarity and effectiveness of Sipser's pedagogical approach, which uses examples of simple programs and reductions to illustrate the halting problem, all stated in the Methodology, are analyzed. Finally, in the Conclusions, the accessibility of the authors' insights, the role in clarifying computational limits, and the value in education are evaluated.

D. Aaronson's Work

Aaronson's (2016) paper, "The Ghost in the Quantum Turing Machine," explored the halting problem in quantum computing.

In the Literature Review of Aaronson's work, the scope of classical and quantum references is reviewed to establish the context for undecidability in new models. Then, the theoretical rigor of the authors' quantum Turing machine definitions that incorporate quantum states, superposition, and entanglement, and the application to the halting problem, all stated in the Methodology, are examined. Finally, in the Conclusions, the novelty of Aaronson's findings, specifically on the parallels to classical undecidability and the relevance to complexity theory, are assessed.

E. Cotler and Rezchikov's Work

Cotler and Rezchikov's (2024) paper applied the halting problem to computational dynamical systems.

In the Literature Review of the authors' work, the integration of classical computability, undecidability, and dynamical systems research is analyzed to frame the study. Then, the proposed combination of computational simulations and theoretical models that explore the halting problem in iterative, chaotic systems with defined assumptions, as stated in the work's Methodology, is critiqued. Finally, in the Conclusions, the applicability of the authors' findings to modern challenges, the interdisciplinary insights, and the contribution to complexity theory are reviewed.

F. Synthesizing the Analyses

The five works are compared to identify overarching themes, contributions, and gaps, thus synthesizing the findings. The analysis explores how Turing's foundational proof establishes the baseline for undecidability, which Pavlotskaya's work extends with mathematical depth and Sipser's work clarifies for broader understanding. Aaronson's work introduces a novel quantum perspective, while Cotler and Rezchikov's work bridges dynamic systems research, highlighting modern relevance. Consistencies, such as the consensus on undecidability, and divergences, such as varying areas of focus like mathematical, educational, quantum, or dynamic, are examined. The strengths of the works are aggregated, including theoretical rigor, accessibility, and innovation, while weaknesses, such as limited practical applications or untested assumptions, are highlighted. This synthesis identifies gaps in practical utility, empirical validation, and interdisciplinary connections, providing a holistic view of the halting problem's role in complexity theory and guiding future research.

IV. RESULTS AND ANALYSIS

Turing's (1937) paper establishes the foundation for undecidability and the halting problem. The literature review, though limited due to the nascent field of computability, references Hilbert's Entscheidungsproblem to frame the halting problem. The methodology formally defines Turing machines, employing diagonalization proof to demonstrate the problem's undecidability, significantly shaping computational complexity theory. The work offers rigorous logic and a clear model but lacks practical focus and assumes an infinite tape, rendering it unrealistic for real-world applications.

Pavlotskaya's (2002) paper advances understanding of undecidability and the halting problem. A robust literature review cites Turing and related works, grounding the study. The methodology provides formal proofs, using axioms for Turing machines (M, inputs w) and step-by-step derivations with precise notations and clear mathematical proofs to establish undecidability. The conclusion reinforces the halting problem's undecidability, linking mathematical proof to formal theory. The paper excels in mathematical rigor and clarity but demands advanced knowledge and lacks real-world applications.

Sipser's (2013) chapter effectively educates undecidability and the halting problem. The literature review offers a broad perspective, citing Turing and others to provide context. The pedagogical methodology uses examples like simple programs and reductions to explain undecidability, ensuring accessibility and clear steps. The conclusion clarifies computation limits, aiding learners in grasping undecidability. The strength lies in its lucid, example-driven approach, though it takes a simplified, broad perspective on the halting problem.

Aaronson's (2016) paper innovates by connecting the halting problem to quantum computing. A comprehensive literature review covers classical, and quantum works, framing undecidability in a new context. The theoretical methodology defines quantum Turing machines (QTMs) with quantum states and superposition, exploring halting in quantum contexts. The conclusion suggests quantum undecidability parallels classical limits. The paper effectively bridges classical theory with modern fields like quantum computing but remains highly speculative with no empirical testing.

Cotler and Rezchikov's (2024) paper links classical works to computational dynamic systems. An extensive literature review integrates Turing's work with undecidability in dynamic systems. The methodology combines simulations and theory, modeling the halting problem in iterative and chaotic systems with discrete states. The conclusion connects the halting problem to complex systems, offering new perspectives for modern and future systems. The interdisciplinary approach is innovative, but the recent work's assumptions remain untested, with limited discussion on scalability.

A comparison of the five papers is presented in Table II, analyzing literature review, methodology, conclusion, strengths, weaknesses, and overall impact on complexity theory.

TABLE II. COMPARATIVE ANALYSIS OF KEY PAPERS

Aspect	**Turing (1937)**	**Pavlotskaya (2002)**	**Sipser (2013)**	**Aaronson (2016)**	**Cotler & Rezchikov (2024)**
Lit. Review	Minimal, cites Hilbert	Robust, cites Turing	Broad, educational	Comprehensive, quantum	Extensive, interdisciplinary
Methodology	Diagonalization, rigorous	Proofs, precise	Examples, clear	QTMs, theoretical	Simulations, innovative
Conclusion	Undecidability proven	Links to mathematical theory	Clarifies limits	Quantum parallels	Relevance to dynamics
Strengths	Foundational, logical	Rigor, clarity	Accessible, examples	Novel, bridges fields	Interdisciplinary
Weaknesses	No practical focus	Less accessible	Simplified	Speculative	Untested assumptions
Impact	Defined computability	Deepened theory	Educated learners	Quantum insights	Moden applications

V. DISCUSSION AND FUTURE DIRECTIONS

The halting problem is a foundational concept in complexity theory, defining the limits of computation. Turing's (1937) proof established its undecidability, showing that no algorithm can determine whether every program halts. Pavlotskaya (2002) deepened this with formal mathematical analysis of Turing machines. Sipser (2013) made the theory accessible through clear educational examples. Aaronson (2016) extended the discussion to quantum computing, exploring undecidability in quantum Turing machines. Cotler and Rezchikov (2024) connected the halting problem to computational dynamical systems, relating it to chaos and iteration.

The halting problem's undecidability reverberates across multiple domains. In computability, it delineates the limits of algorithmic solutions, establishing boundaries for what can be computed. In complexity theory, it shapes problem classification, distinguishing decidable from undecidable problems and guiding research into tractable solutions. Software verification faces significant challenges, as no general tool can automatically determine program termination, critical for ensuring reliability and safety in software development. In algorithm design, it influences strategies, pushing focus toward efficient, approximate, or domain-specific solutions. Quantum computing, as explored by Aaronson, raises questions about undecidability in new computational paradigms, potentially redefining limits. Dynamical systems, per Cotler and Rezchikov, connect the problem to real-world phenomena like chaos in physics, biology, and engineering, highlighting its broad relevance. In AI safety, the inability to predict program behavior complicates efforts to ensure robust, predictable systems, a growing concern in machine learning and autonomous technologies.

The reviewed works reveal persistent challenges. The theoretical focus of Turing and Pavlotskaya limits direct applications, leaving a gap between theory and practice. Sipser's simplification, while educational, lacks the depth for advanced research. Aaronson's speculative quantum models require empirical validation to confirm their feasibility, while Cotler and Rezchikov's untested assumptions about chaotic systems demand further exploration for scalability. These gaps present opportunities for bridging theory to practice, validating new models, and extending the halting problem's insights into emerging fields. The consensus on undecidability underscores a unified theoretical foundation with divertive approaches, thus suggesting potential for integration and innovation.

Future research directions should aim to address these challenges and leverage opportunities, including, but not limited to:

A. Approximation Algorithms

Develop heuristics to predict halting behavior in restricted scenarios, such as programs with bounded loops, finite inputs, or specific structures. These could enhance software testing, debugging, and verification tools, offering practical solutions despite theoretical limits.

B. Quantum Computing

Build on Aaronson's work to explore quantum Turing machines and algorithms and investigate how quantum properties like superposition and entanglement might address undecidability and potentially uncover new computational boundaries or partial solutions.

C. Dynamical Systems

Extend Cotler and Rezchikov's models, applying halting analysis to real-world systems like weather prediction, neural networks, or biological processes. This could yield insights into termination and stability in chaotic, iterative contexts, thus bridging theory to practice.

D. Practical Tools

Design frameworks for approximate halting detection, tailored for industry use. These tools could balance theoretical constraints with usable outcomes, aiding developers in software reliability and performance optimization.

E. Empirical Validation

Test assumptions in Aaronson and Cotler and Rezchikov's works through experiments or computational simulations. Validate quantum models with prototype quantum systems and dynamic models with real-world data, strengthening their credibility and applicability

F. Artificial Intelligence and Safety Applications

Investigate the halting problem's role in AI safety, developing methods to approximate behavior in machine learning models. This could address risks in autonomous systems, ensuring safer, more predictable outcomes.

ACKNOWLEDGEMENT

Acknowledgment to the Ministry of Higher Education Malaysia (MOHE) through Fundamental Research Grant Scheme (FRGS) Ref: FRGS/1/2024/ICT02/UTM/02/10, Vot. No: R.J130000.7828.5F748.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *s2-42*(1), 230–265. https://doi.org/10.1112/plms/s2-42.1.230.

[2] Pavlotskaya, L. M. (2002). Turing machines connected to the undecidability of the halting problem. *Mathematical Notes, 71*, 667–675. https://doi.org/10.1023/A:1015840005656.

[3] Sipser, M. (2012). *Introduction to the theory of computation (3rd ed.)*. Cengage Learning.

[4] Aaronson, S. (2016). The ghost in the quantum turing machine. *The Once and Future Turing*, 193–296. https://doi.org/10.1017/cbo9780511863196.018.

[5] Cotler, J., & Rezchikov, S. (2024). Computational dynamical systems. In *Proceedings of the 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS 2024)* (pp. 166–202). IEEE Computer Society. https://doi.org/10.1109/FOCS61266.2024.00021.