

Article

Not peer-reviewed version

Logical-Time Incompleteness: A Structural Boundary of Artificial Intelligence

[Hongliang Shen](#) *

Posted Date: 3 March 2026

doi: 10.20944/preprints202603.0085.v1

Keywords: logical time; task completion; incompleteness; artificial intelligence; realization; task structure



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Logical-Time Incompleteness: A Structural Boundary of Artificial Intelligence

Hongliang Shen

Independent Researcher, USA; shl.1985@tsinghua.org.cn

Abstract

This paper examines the limits of artificial intelligence by distinguishing two dimensions that are frequently conflated in contemporary discussions: *machine realization* and *task completion*. While advances in AI are often understood as improvements in realization—better architectures, larger models, or more efficient learning—such advances do not necessarily imply that a task, as specified, is completable at all. The paper argues that this assumption obscures a class of structural limits that are independent of computational power or intelligence. To make this distinction precise, a task-relative framework based on *logical time* is introduced. Logical time characterizes the organization of conceptually required transitions leading toward a task's completion conditions, abstracting away from implementation details. Within this framework, the notion of *logical-time incompleteness* is defined as a property of tasks whose completion conditions cannot be reached along any logically bounded trajectory. Several general structural sources of incompleteness are identified. On this basis, the paper formulates the **AI Boundary Thesis**: if a task is logically-time incomplete under its own specification, then no machine—regardless of realization—can fully complete it as specified. This perspective helps explain persistent phenomena in AI, including reliance on approximation, the appearance of pseudo-completion, and the contrast between superhuman performance on formal tasks and fragility in everyday reasoning.

Keywords: logical time; task completion; incompleteness; artificial intelligence; realization; task structure

1. Introduction

The question of whether artificial intelligence has fundamental limits, and how such limits ought to be articulated, has accompanied the field since its inception. The rapid and unexpected performance gains of recent AI systems, particularly large language models, have intensified both the excitement and the uncertainty surrounding the boundaries of machine capabilities. What was once largely a theoretical concern has become a matter of practical and social significance. Without a framework that can diagnose the nature of AI limitations, we risk overestimating the potential of purely computational improvements.

At the same time, these advances have further accentuated a broad, Moravec-like paradox (Moravec 1988):

Artificial intelligence has achieved superhuman proficiency in domains traditionally associated with high-level intelligence, such as strategic games, formal reasoning, and scientific optimization. At the same time, AI systems continue to struggle with tasks that appear trivial for humans, including casual conversation, social reasoning, and commonsense understanding. This contrast has been widely discussed, yet remains insufficiently explained at a structural level.

This divergence suggests that the conventional ways of characterizing task difficulty, computational complexity, representational demands, or informational uncertainty, are insufficient for explaining the observed landscape of AI strengths and weaknesses.

From a more general perspective, the viewpoints adopted in existing research and discussions make it particularly difficult to distinguish between two questions that are fundamentally distinct: how a machine completes a task, and whether the task is completable at all.

This paper proposes a new perspective: a task-relative framework that addresses this question by introducing the notion of **logical time**. Logical time abstracts away from physical duration and computational cost, focusing instead on the structure and organization of conceptually necessary transitions required to reach a task's completion conditions. Within this framework, tasks are analyzed not in terms of how they are implemented, but in terms of whether their own specifications permit termination along any admissible conceptual trajectory.

The central concept developed here is **logical-time incompleteness**. A task is logically-time incomplete if no permissible conceptual evolution—finite, infinite, or transfinite—ever reaches a state that satisfies the task's completion conditions. Such incompleteness is not a consequence of limited resources, insufficient intelligence, or inadequate learning, but a structural property of the task itself.

On this basis, the paper advances a category distinction that is frequently blurred in discussions of artificial intelligence: the distinction between **machine realization** and **task completion**. Research on AI realization concerns how cognitive or computational processes are implemented, optimized, or approximated by physical or algorithmic systems. Task completion, by contrast, concerns whether the task's own completion conditions are reachable at all. These two dimensions are orthogonal. Improvements in realization do not, by themselves, alter the completion status of logically-time incomplete tasks.

Clarifying this distinction enables a structural explanation of several familiar phenomena: superhuman performance on formally well-structured tasks, persistent difficulty with seemingly elementary human activities, the central role of approximation in successful AI systems, and the emergence of pseudo-completion and pseudo-determinism in high-stakes applications. These phenomena are not merely contingent technological shortcomings, but consequences of the logical-time structure of the tasks to which AI systems are applied.

Taken together, the paper makes three interrelated contributions.

First, the paper clarifies a category distinction that has remained largely implicit and frequently conflated in discussions of artificial intelligence: the distinction between machine realization and task completion.

Much of the contemporary debate assumes that advances in realization, better architectures, more data, or greater computational power, necessarily advance completion. The paper argues that this assumption is structurally mistaken. Realization concerns how a task is implemented or approximated, whereas completion concerns whether a task, as specified by its own conceptual structure, admits termination at all.

Second, to make this distinction precise, the paper introduces a task-relative framework based on logical time.

Tasks are characterized in terms of conceptual state spaces, admissible transitions, and completion conditions. Logical time does not measure physical duration or computational effort, but the structural organization of conceptually necessary stages required for task completion. Within this framework, the notion of logical-time incompleteness is defined as a property of tasks whose completion conditions are structurally unreachable along any bounded logical-time trajectory.

Third, on this basis, the paper formulates the AI Boundary Thesis: if a task is logically-time incomplete under its own specification, then no machine, regardless of architecture, computational power, or representational format, can fully complete the task as specified.

This boundary is orthogonal to classical limits such as undecidability, intractability, or learnability. It does not concern what can be computed or learned in principle, but whether a task admits completion at all, given its logical-time structure.

Roadmap

The paper unfolds in three stages. Sections 2–4 develop the task-relative framework by first introducing the notion of logical time, which provides the structural basis for analyzing completion conditions, approximation, and the possibility of logical-time incompleteness. Building on this foundation, Sections 5 and 6 articulate the paper’s core theoretical claims: a threefold classification of tasks, the AI Boundary Thesis, and a principled distinction between machine realization and task completion. Section 7 discusses implications for contemporary AI systems and related philosophical issues. Section 8 concludes.

2. A Task-Relative Framework and the Notion of Logical Time

This section introduces the task-relative framework within which the subsequent analysis is conducted. The purpose of the framework is not to model cognitive or computational processes as such, but to characterize tasks in a way that makes their completion conditions and structural limits explicit. In particular, it provides the basis for distinguishing questions of *how* a task is realized from questions of *whether* the task admits completion at all.

2.1. Tasks, State Spaces, and Completion Conditions

A task is specified as a triple

$$T = (LS, R, C),$$

where:

- LS is a space of conceptual states. By “conceptual state”, we mean a task-relative position within the logical-time structure of the task, not a psychological or representational state of any agent. These represent the cognitive logical positions that an agent or system may occupy while attempting to perform the task.
- $R \subseteq LS \times LS$, is the conceptual transition relation. An ordered pair (s, t) indicates that moving from state s to state t is a permissible conceptual step in attempting to complete the task. No assumptions are made about the determinacy, functionality, or well-foundedness of R ; transitions may branch, cycle, or form infinite descending chains.
- $C \subseteq LS$ is the completion set, consisting of those states that satisfy the task’s completion conditions. The set C may be empty or nonempty and need not be reachable from any particular initial state. Crucially, the nonemptiness of C is a purely structural property of the task specification: it indicates the existence of states that would count as completion states if reached, but it does not imply that the task has been, can be, or will ever be completed by any system.

Intuitively, LS represents the possible conceptual configurations relevant to the task, R encodes which conceptual transitions are admissible within the task’s specification, and C specifies the condition under which the task is considered complete.

This task-relative formulation abstracts away from any particular agent, algorithm, or physical system. It does not presuppose symbolic, statistical, neural, or hybrid mechanisms, nor does it assume any specific resource bounds. Instead, it treats tasks as conceptually specified structures whose completion conditions are defined independently of their realization.

Completion, in this sense, is evaluated relative to the task’s own specification. A task is completed only if a conceptual state satisfying the completion condition is reached through admissible transitions. Whether such a state is reachable, and under what structural constraints, is the central concern of the present framework.

2.2. Logical Time

Logical time characterizes the ordered stages of conceptual progression induced by the transition structure R . A *logical trajectory* is a sequence

$$\tau = (LS_0, LS_1, LS_2, \dots)$$

such that $(LS_i, LS_{i+1}) \in R$ for all i . The index of each LS_i is interpreted as a logical-time step: stage 0 represents the initial conceptual state, stage 1 the next conceptual refinement, and so on. Logical time thus measures the *depth of conceptual progression* required by the structure of R , rather than the duration of any physical or computational process.

The formalism permits logical trajectories of arbitrary ordinal length. Given an ordinal α , a trajectory of length α is a function

$$\tau : \alpha \rightarrow LS,$$

such that for all $\beta < \alpha - 1$,

$$(\tau(\beta), \tau(\beta + 1)) \in R.$$

It is important to emphasize that logical time, as defined here, is *not* identified with any particular sequence of logical states. A sequence such as $\tau = (LS_0, LS_1, LS_2, \dots)$ represents a logical-time trajectory—one admissible realization of the task's logical structure.

Logical time may be *well-founded*, in which case every admissible trajectory has a maximal length, or *non-well-founded*, in which case trajectories may descend indefinitely without reaching a terminal point. Both cases are essential for analyzing the structural limits of task completion.

Example 1 Consider the simplest computational task “ $1+1 = ?$ ”. In terms of logical time, it can be expressed as follows:

- LS_0 : Seeing the expression “ $1+1 = ?$ ”, without having performed any conceptual operation yet.
- LS_1 : Recognizing that this is an addition task.
- LS_2 : Initiating the addition algorithm for “ $1+1$ ”.
- LS_3 : Reaching the conclusion “ 2 ”.
- $LS\checkmark$: Filling in the answer “ 2 ” and completing the task.

Thus:

$$\tau = \{LS_0, LS_1, LS_2, LS_3, LS\checkmark\}$$

Here one can clearly observe the difference between the dimension examined by logical time and the dimension examined by traditional approaches. More specifically, computational complexity concerns what happens inside the second step.

2.3. Relation to Existing Approaches

The task-relative framework and the notion of logical time intersect with several existing lines of research, but they are not reducible to any of them. In each case, the difference is not one of theoretical rivalry, but of the dimension along which tasks are characterized.

First, the present framework is orthogonal to classical limits in computation theory, such as undecidability, intractability, or resource-bounded complexity (Gödel 1931; Turing 1936; Sipser 2013). These limits characterize what can be computed, decided, or optimized by idealized machines under specified assumptions, and thus provide measures of computational feasibility or efficiency. Logical time, by contrast, introduces a structural metric on tasks themselves: it concerns whether a task's completion conditions are reachable along any logically bounded conceptual trajectory. This question is prior to, and independent of, considerations of computational power, algorithms, or resource constraints.

Second, the framework is distinct from theories that focus on cognitive complexity (e.g., Newell and Simon, 1976) or representational richness (e.g., Fodor, 1975). Such approaches offer measures of

how demanding a task is to realize, in terms of search, resources, or expressive capacity. Logical time likewise functions as a form of measurement, but along a different dimension. Rather than quantifying degrees of difficulty, it characterizes whether a task admits a determinate notion of completion at all. The resulting measures are therefore orthogonal: they address different aspects of task structure and do not compete to explain the same phenomenon.

Third, although the present framework is superficially reminiscent of work in dynamic semantics, game-theoretic reasoning, and the study of non-well-founded or circular structures (e.g., Kamp 1981; van Benthem 1996; Aczel 1988), no claim of theoretical integration, extension, or unification is intended. Those approaches investigate how reasoning proceeds, how information is updated, or how interactive and circular processes are modeled. By contrast, the framework proposed here does not provide a model of reasoning, interpretation, or interaction.

Instead, it isolates a strictly prior and structurally simpler question: whether a task, as specified by its own completion conditions, admits any position at which a process is permitted to terminate. Logical time is introduced solely to express this reachability condition. No assumptions are made about internal mechanisms, representational resources, or inferential dynamics. For this reason, the present framework neither competes with nor subsumes existing approaches. It remains compatible with multiple—and even mutually incompatible—accounts of reasoning or interpretation, since it operates at a different logical type: it asks not how a process unfolds, but whether the task structure allows the process to end at all.

3. Completion, and Approximation Completion

3.1. Logical-Time Completion

A trajectory τ completes the task if there exists an ordinal β such that $\tau(\beta) \in C$. If such a β exists, the completion time of τ is the least such β . When no such β exists, the trajectory is non-completing.

A task is logically completable if there exists at least one trajectory that reaches a completion state:

$$\exists \tau \exists \beta (\tau(\beta) \in C).$$

Prose explanation. Completion occurs when some step in the reasoning process lands in a state designated as sufficient for success. This captures a wide range of phenomena: solving a puzzle, reaching a belief state that resolves uncertainty, or arriving at a linguistic interpretation. Crucially, completion is not assumed to be unique, immediate, or computationally detectable; it is simply the attainment of a structurally appropriate endpoint.

3.2. Partial and Approximate Completion

The preceding discussion has treated completion as a determinate terminal condition. However, this assumption is not forced by the framework itself. In many domains, it is often unclear what would count as a precise or final completion state. In practice, humans routinely accept approximate resolutions to tasks that are logically incompletable. For example, in natural language interpretation, speakers do not resolve all semantic ambiguities; in social reasoning, agents do not iterate indefinitely through higher-order belief attributions (“I believe that you believe that I believe...”); and in negotiation, partial agreement is often sufficient for successful coordination.

This perspective echoes Herbert Simon’s notion of bounded rationality, where satisficing replaces strict optimization (Simon 1957). To capture this structural feature, the framework allows an approximation sequence, a chain

$$Ls_0 \leq Ls_1 \leq Ls_2 \leq \dots$$

generated by a preorder $\leq$ interpreting increasing adequacy. A sequence approximates completion if it is increasing and unbounded, even if it lacks a maximal element.

Approximate completion explains how agents, particularly humans, can successfully perform tasks that are formally non-terminating. By relaxing completion conditions or tolerating residual indeterminacy, agents effectively treat certain trajectories as sufficient for practical purposes.

Machines, by contrast, are often designed to satisfy explicit completion predicates. When such predicates are unattainable, systems may enter indefinite refinement loops or exhibit unstable behavior. Conversely, the success of machines is precisely because people can help them avoid this, as exemplified by the success of LLM.

3.3. Logical Time and the Structural Difficulty of Task Completion

Logical time, as introduced here, measures not physical duration but the number and organization of conceptually necessary transitions required to reach completion. A general structural tendency follows: **tasks that require fewer distinct conceptual stages to reach their completion conditions admit fewer points at which delay, error, or coordination failure may arise.**

Reducing logical-time length does not guarantee faster execution, nor does it presuppose optimal performance. Rather, it reduces the structural opportunities for failure by eliminating intermediate dependencies, interpretive steps, or condition checks that must otherwise be successfully coordinated. In this task-relative sense, shortening logical time increases the **structural ease of completion.**

This perspective explains why tasks that appear similar at the level of outcome may differ markedly in their difficulty of completion, and why experience, abstraction, or task reformulation often lowers difficulty by collapsing multiple conceptual stages into a single logical transition. The claim is not empirical but structural: difficulty tracks the organization of logical time required for completion, not factors such as intelligence, skill, or execution strategy.

Example 2

Consider the assembly of two identical bookcases by the same person. Let τ_1 denote the task of assembling the first bookcase, and τ_2 the task of assembling the second.

For τ_1 , the logical-time progression may be described as follows:

- **LS₀**: unpacking the materials and locating the instruction manual;
- **LS₁**: reading and understanding the instructions;
- **LS₂**: assembling the bookcase step by step according to the instructions;
- **LS_✓**: task completion.

For τ_2 , the logical-time progression is shorter:

- **LS₀**: unpacking the materials and locating the instruction manual;
- **LS₂**: assembling the bookcase according to the already understood procedure;
- **LS_✓**: task completion.

The second task omits the conceptual stage of understanding the instructions. As a result, the logical-time progression is shorter. This illustrates how experience or familiarity effectively reduces the number of conceptually necessary stages, thereby lowering the structural difficulty of task completion.

Example 3

As discussed in Example 1 of Section 3.2.1, consider the same arithmetic task when performed by a child who is unfamiliar with arithmetic. In this case, the logical-time progression can be informally described as follows:

- **LS₀**: seeing the expression “ $1 + 1 = ?$ ”, without having performed any conceptual operation;
- **LS₁**: not recognizing the task and being informed that it is an addition problem;

- LS_1' : learning the addition algorithm
- LS_2 : initiating the addition algorithm for “1 + 1”;
- LS_3 : reaching the conclusion “2”;
- $LS\checkmark$: filling in the answer “2” and completing the task.

Thus, the logical-time progression for this agent involves additional conceptual stages:

$$\tau = \{LS_0, LS_1, LS_1', LS_2, LS_3, LS\checkmark\}$$

The inclusion of these extra conceptual stages lengthens the logical-time progression and thereby increases the structural difficulty of completing the task.

Taken together, these examples support a minimal structural observation: task difficulty correlates positively with the length and organization of the logical-time progression required for completion.

4. Logical-Time Incompleteness

Logical-time incompleteness is the central notion of this paper. It captures the idea that certain tasks cannot, even in principle, be brought to completion because the conceptual evolution required by the task never reaches a terminal state. This section provides a precise definition, identifies general structural sources of incompleteness.

4.1. Definition (Logical-Time Incompleteness).

A task $T=(LS,R,C)$ is **logically-time incomplete** if and only if:

$$\forall \tau \forall \beta (\tau(\beta) \notin C),$$

that is, for every admissible logical-time trajectory τ , no stage of the trajectory ever reaches a completion state.

Equivalently, the completion set C is structurally inaccessible under the transition relation R . No permissible conceptual evolution, no matter how long, even if extended into the transfinite, reaches a state at which the task's own conditions for completion are satisfied.

Logical-time incompleteness is therefore a **structural property of the task itself**. It does not arise from the limitations of particular agents, algorithms, or architectures, but from the topology of the conceptual space and the accessibility relations defined by the task specification.

4.2. A Minimal Toy Model

To illustrate logical-time incompleteness, consider a minimal toy model of linguistic disambiguation.

Let

$$\tau = \{LS_0, LS_1, LS_2, \dots\}$$

where each state LS_n represents an interpretive hypothesis refined to degree n .

Let $R \subseteq LS \times LS$ be a refinement relation such that

$$(LS_n, LS_{n+1}) \in R,$$

capturing a single step of conceptual or contextual refinement.

Let the completion set be

$$C = \emptyset.$$

This reflects the claim that, due to pragmatic and contextual underdetermination, there exists no state in which all semantic ambiguity is fully resolved.

This assumption is not an ad hoc simplification of the model, but reflects a structural limitation of semantic interpretation itself. In particular, the full elimination of semantic ambiguity does not admit a closed completion condition. Related limits have been independently identified across several research traditions. In philosophy of language, arguments from the indeterminacy of

translation and radical interpretation show that even under idealized conditions, meaning cannot be uniquely fixed by any finite set of linguistic or behavioral data (Quine 1960; Davidson 1973). In pragmatics, it has been argued that context-sensitive enrichment is integral to truth-conditional content, such that attempts at disambiguation systematically introduce new implicit assumptions rather than exhaust the interpretive space (Grice 1975; Recanati 2004). From the perspective of artificial intelligence and cognitive science, the symbol grounding problem and work on commonsense reasoning suggest that semantic determination cannot be achieved within a purely formal, closed symbolic system (Harnad 1990; Hayes and Ford 1995). Within the present framework, these dispersed limitations are captured as a failure of logical-time completion: the task of semantic disambiguation does not admit a closed completion condition.

The details are as follows:

Linguistic interpretation typically involves iterative resolution of multiple sources of indeterminacy, including lexical ambiguity, pragmatic enrichment, contextual underspecification, vague predicates, presuppositions, conversational implicatures, and discourse coherence constraints. Each refinement step corresponds to a transition in R , and a trajectory in logical time takes the form $\tau = \langle L_{s_0}, L_{s_1}, L_{s_2}, \dots \rangle$, indexed by ω , where later states encode increasingly specific interpretive commitments.

Crucially, many forms of linguistic refinement lack a terminal state. Vague predicates admit no maximally precise interpretation; presupposition resolution may trigger further presuppositions indefinitely; contextual disambiguation can generate unbounded chains of inferential commitments; and conversational implicatures dynamically adjust to higher-order expectations of rationality. As a result, no trajectory reaches a state in C , since the completion condition corresponding to a fully determined meaning is not well defined.

The task is therefore logically incomplete in logical time: interpretive refinement proceeds indefinitely without convergence to a terminal state. If a preorder is introduced on LS to represent increasing interpretive adequacy, trajectories may approximate completion arbitrarily well without attaining it. Human interpreters may accept such sufficiently refined states as practically adequate, whereas systems requiring strict completion cannot. This gap between approximation and completion will play a central role in the subsequent analysis.

4.3. Structural Sources of Incompleteness

Logical-time incompleteness can arise from several general structural configurations. The following sources are not tied to specific cognitive theories or computational models; they recur across domains such as linguistics, game theory, social reasoning, and interactive computation.

(i) Non-well-founded or infinite-descending transitions.

If the transition relation contains cycles or infinite descending chains, trajectories may extend indefinitely without reaching completion.

(ii) Empty completion sets.

In tasks involving open-ended refinement or principled underdetermination, the completion set may be empty, rendering incompleteness immediate.

(iii) Completion sets outside the reachability region.

If C lies entirely outside the closure of the reachable region of R , no trajectory ever reaches C , even if C itself is nonempty.

(iv) Shifting or unstable completion conditions.

In multi-agent settings, the completion predicate may itself depend on higher-order reasoning that changes as conceptual states evolve. This yields a moving-target problem where no stable completion state exists.

(v) Reflexive or mutually recursive update structures.

Iterated belief reasoning and self-referential updates ("I think that you think that I think...") may generate infinite ascent in logical time without fixed points.

These structural sources identify incompleteness as a property of task design rather than agent capacity. They prepare the ground for the task classification and boundary thesis developed in the next section.

5. Logical Time and AI Tasks

5.1. Several Typical Tasks in AI

This section applies the framework to several domains in which contemporary AI systems exhibit divergent performance. The aim is not to provide empirical measurements, but to show how logical time offers a principled structural explanation for observed behavioral patterns. Each case illustrates a different manifestation of how task topology determines whether full completion is attainable.

5.1.1. Perfect-Information Games: Logical-Time Completion

In perfect-information games such as Go, Checkers, and certain forms of Chess endgames, the structure of the task is represented by a well-founded game tree. The transition relation

R consists of legal moves, and conceptual states correspond to positions or evaluation hypotheses. Because such games are finite (or finitely branching with limited depth), every trajectory eventually terminates in a terminal position. Thus, regardless of how large the state space may be, the logical time structure is well-founded.

In this setting, the completion set C consists of states satisfying the goal condition (e.g., win, loss, draw). Since every legal sequence of moves reaches a terminal position, the task is logically completable. Logical-time incompleteness does not arise, because no conceptual refinement can extend indefinitely: once all legal moves are exhausted, the game ends.

This structural property explains the well-known success of AI systems in perfect-information games. The logical-time topology itself places no structural barrier on machine realizability.

5.1.2. Incomplete-Information Games: Conditional Completion and Structural Instability

Games such as Poker, Bridge, or auction settings introduce incomplete information and higher-order uncertainty about other agents' strategies and beliefs. Here, conceptual states incorporate beliefs about hidden information and expectations about others' reasoning. The transition relation R involves updates in belief states or strategy hypotheses.

Unlike perfect-information games, these belief-update dynamics sometimes do not stabilize. Higher-order belief modeling ("I think that you think that I think...") produces trajectories that may ascend indefinitely in epistemic depth. Moreover, optimal strategies in such settings depend on beliefs about others' adaptive strategies, leading to strategic cycles rather than fixed points. So completion conditions C (optimal or equilibrium strategies) may be structurally unreachable.

Thus, incomplete-information games frequently instantiate logical-time incompleteness. Even though humans can act effectively by adopting approximate equilibria or pragmatic heuristics, machines designed around strict completion criteria may struggle with tasks whose conceptual structure lacks any terminating trajectory.

This difference in logical structure helps explain why AI performance in Poker lags behind its achievements in Go, despite Poker being computationally simpler in several senses.

5.1.3. Social and Rule-Shifting Games: Open-Ended and Non-Well-Founded Structure

Tasks such as Mahjong, informal negotiation games, or multiplayer role-based games exhibit even more complex forms of incompleteness. Their conceptual structure involves:

- tacit norms;
- unstable or implicitly negotiated rules;
- shifting coalitions;

- contextually contingent interpretations of moves;
- higher-order social expectations.

The transition relation R in such games may involve non-well-founded cycles: a conceptual state representing a social expectation can require further expectations (“I expect that you expect that I will conform to the rule that you believe others expect...”), generating patterns that lack any terminal state.

The completion predicate C in these settings is typically defined in terms of socially recognized “successful play,” but such states may be pragmatic, approximate, or context-relative, not fixed structural endpoints. As a result, the notion of “full completion” may be inappropriate for capturing human performance.

Machines trained on statistical regularities may approximate human behavior in restricted domains, but cannot fully realize tasks whose completion conditions are structurally ungrounded. Logical-time incompleteness explains this gap: the tasks themselves disallow terminating conceptual trajectories.

This structural difference sheds light on the persistent difficulty AI systems face when required to adhere to socially negotiated norms that adapt dynamically to human expectations.

In addition to the examples above, as mentioned earlier, natural language understanding provides a paradigmatic example of logical-time incompleteness; it will not be repeated here.

5.2. Logical-Time-Based Classification of AI Tasks

Taken together, the preceding analysis supports a principled classification of AI tasks based on their behavior in logical time. From this perspective, tasks differ not merely in computational complexity or data requirements, but in the structural properties of their conceptual trajectories. In particular, tasks can be divided into three broad classes according to whether and how their logical-time processes converge.

Class I tasks are characterized by **finite logical-time convergence**. For such tasks, every admissible trajectory reaches a completion state after a bounded number of conceptual steps. The completion condition is well-defined and structurally accessible. As a result, these tasks are fully realizable by machines, and artificial systems can in principle achieve perfect or superhuman performance. Canonical examples include perfect-information games such as Go and formally specified domains such as arithmetic proof.

Class II tasks exhibit asymptotic logical-time convergence. In these cases, conceptual refinement proceeds indefinitely, but trajectories approach a stable region or limit in an increasingly constrained manner. Although no final completion state may be reached in a strict sense, partial or approximate completion is well-defined, and the residual error can be bounded or controlled. Tasks of this kind are therefore not fully completable, but they are *approximately realizable*. Strategic games with incomplete information, such as Mahjong or Poker, as well as certain predictive tasks like stock market forecasting, fall into this category.

Class III tasks display divergent or cyclic logical-time behavior. Their conceptual trajectories neither terminate nor converge, but instead exhibit unbounded refinement, oscillation, or structural instability. In these tasks, no completion condition is structurally reachable, even in the limit. Consequently, they possess an inherent logical-time incompleteness. Artificial systems may generate plausible outputs or local approximations, but full task realization is structurally precluded. Paradigmatic examples include semantic disambiguation in natural language, where meaning refinement can proceed indefinitely without a determinate endpoint.

Table 1. Logical-Time-Based Classification of AI Tasks.

Task Class	Structural Property	Logical-Time Completeness	AI Performance	Representative Examples
------------	---------------------	---------------------------	----------------	-------------------------

Class I	Finite convergence	Completable	Fully realizable; perfect solutions possible	Go, arithmetic proof
Class II	Asymptotic convergence	Partially completable	Approximable with bounded error	Mahjong, Poker, stock prediction
Class III	Divergent or cyclic	Incompletable	Structurally unrealizable	Semantic disambiguation

This threefold classification highlights that the limits of artificial intelligence arise not primarily from insufficient computational power or learning capacity, but from the logical-time structure of the tasks themselves. It is important to emphasize that this taxonomy is **structural rather than empirical**. The classification does not depend on current technological limitations, training regimes, or architectural choices, but on the existence and accessibility of completion conditions within the task specification.

Logical time thus functions as a unifying dimension that explains why certain tasks admit full mechanization, others admit only approximation, and still others resist completion altogether. At the same time, this framework offers a clear resolution to a long-standing puzzle in the philosophy of AI: why machines often surpass humans in tasks that appear to require extraordinary intelligence (e.g., Go, chess, molecular design), yet struggle with tasks that require only basic human competence (e.g., casual conversation or social reasoning).

The apparent paradox dissolves once logical time is taken into account. Tasks in which machines achieve superhuman performance may have relatively simple logical-time structures, even when they are computationally demanding. Conversely, tasks that humans perform effortlessly may involve complex, open-ended, or non-terminating logical-time structures, despite their minimal subjective cognitive load. Machine intelligence therefore aligns with logically time-completable tasks, rather than with “high-intelligence” tasks per se.

This leads to the following **pre-thesis**, stated without the ambition of formal proof.

Pre-Thesis (Task Completion and Logical Time). In artificial intelligence, the difficulty of completing a task is not primarily determined by the apparent cognitive sophistication of the operations involved, but by the structure of the task’s logical-time progression. Tasks that admit completion only along long or unbounded logical-time trajectories are proportionally more difficult to complete than tasks whose logical-time requirements are short and well bounded.

What matters, from this perspective, is not whether a task involves deep reasoning, rich representations, or complex internal computations, but whether its completion conditions can be satisfied along a logical-time trajectory whose length is structurally constrained. This observation provides the conceptual bridge to the boundary thesis developed in the next section.

6. The AI Boundary Thesis

This section articulates the central claim of the paper: a structural boundary on machine realizability that arises from the topology of task completion rather than from limitations of computation, representation, or learning. The claim is not presented as a formal theorem within a fixed axiomatic system, but as a general thesis supported by the task framework and the notion of logical-time incompleteness developed in the preceding sections.

6.1. A Category Clarification: Realization and Completion

Before proceeding, it is necessary to clarify a distinction that plays a central role in the present framework and that has often remained implicit in discussions of artificial intelligence. Much of the traditional literature on AI is concerned with *machine realization*: the problem of how cognitive or intelligent behavior can be implemented in physical or computational systems. Questions about architectures, representations, learning algorithms, efficiency, and scalability all belong to this dimension. Improvements along this axis address how well a task can be *realized* by increasingly sophisticated machines.

The analysis developed here operates along a different, orthogonal dimension, which concerns *task completion*. Completion is not a matter of implementation quality, computational power, or learning capacity, but of whether a task, as specified by its own conceptual structure and completion conditions, admits a trajectory that reaches completion at all. A task may admit arbitrarily advanced realizations and yet remain logically-time incompletionable. In such cases, further progress in realization cannot convert partial or asymptotic performance into genuine completion.

This distinction matters because much of the contemporary debate about the limits, risks, and failures of AI implicitly assumes that better realization necessarily advances completion. This assumption is structurally mistaken. When tasks are logically-time incompletionable, no amount of architectural sophistication, computational scale, or learning capacity can transform them into fully completionable tasks without altering the task specification itself.

A particularly important consequence of conflating realization and completion is the phenomenon of **pseudo-completion**. Pseudo-completion occurs when human users or institutions take a task to be completed, even though the task is logically-time incompletionable and no completion state has in fact been reached. The appearance of completion arises not from the task structure, but from pragmatic acceptance, social trust, or overinterpretation of fluent behavior.

Pseudo-completion often gives rise to **pseudo-determinism**: outputs are interpreted as definitive, authoritative, or truth-tracking, even though they are produced within an approximation regime that structurally tolerates error. This is especially salient in domains such as natural language understanding, social reasoning, ethical judgment, or normative decision-making, where completion conditions are unstable or underdetermined.

The framework developed in this paper isolates this source of limitation. Logical-time incompletionability identifies tasks whose completion conditions are structurally inaccessible, regardless of how well the task is realized. In such cases, what improves over time is not completion but approximation: systems become better at producing outputs that are pragmatically acceptable, statistically reliable, or contextually useful, without ever reaching a genuine completion state.

Recognizing this distinction clarifies why certain failures of AI persist even as systems improve, and why apparent success may coexist with systematic error, instability, or hallucination.

Accordingly, the arguments that follow concern this second dimension, completion rather than realization, and the structural boundaries imposed by logical-time incompletionability.

6.2. Statement of the Thesis

We can now state the central claim.

The AI Boundary Thesis

If a task $T=(LS,R,C)$ is logically-time incompletionable, so that no admissible conceptual trajectory ever reaches a completion state, then no machine can fully complete the task under its specification.

More precisely: whenever the structure of a task requires termination in a completion state that is not reachable in logical time, Machine Completion fails independently of computational power, representational richness, or learning capacity.

6.3. Structural Argument

The boundary thesis can now be articulated in structural terms. The argument does not proceed by formal proof, nor does it rely on a detailed account of machine implementation. Instead, it highlights a mismatch between task specifications and the structure of logical-time progression.

As discussed in Section 5.2, task completion effort correlates positively with the length and structure of the logical-time progression required for completion. Tasks whose completion requires longer or more elaborate conceptual trajectories tend, *ceteris paribus*, to demand greater time and effort. This correlation applies equally to human agents and artificial systems, insofar as both must traverse the conceptual stages specified by the task.

Consider now a task whose logical-time structure admits no completion state. By definition, such a task does not allow any admissible conceptual trajectory to reach a point at which the task's completion condition is satisfied. From a structural perspective, this means that the task requires a form of completion that its own conceptual dynamics cannot deliver.

In such cases, speaking of "task completion" ceases to be meaningful. No amount of computational speed, representational sophistication, or implementation ingenuity can compensate for the absence of a terminating conceptual trajectory. The obstacle lies not in the executing system, but in the task specification itself.

This structural incompatibility explains why tasks that are logically-time incompletionable cannot be completed by machines. The claim does not depend on assumptions about machine architecture or internal state representation. It follows from the simple fact that completion presupposes the existence of a logically admissible endpoint, and that such an endpoint is absent in logically-time incompletionable tasks.

The boundary identified here is therefore not a contingent limitation of artificial systems, but a principled constraint imposed by task structure. It is this constraint that underlies the boundary thesis formulated in the previous section.

6.4. Interpretation

The AI Boundary Thesis identifies a limit that is structural rather than computational. It does not depend on whether machines are fast enough, powerful enough, or trained on sufficient data. Instead, it depends on whether the task specification admits a terminating conceptual pathway.

If a task requires a form of completion that the task's own conceptual dynamics cannot deliver, then no machine can satisfy that requirement. This holds even if machines are allowed unbounded computation, transfinite reasoning steps, or arbitrarily expressive representations.

Accordingly, logical-time incompletionability identifies a structural boundary on task completion rather than a limitation on computation, learning, or intelligence *per se*. This boundary persists even under idealized assumptions about realization, since it arises from the task's own structure rather than from the properties of any particular machine. The framework thus offers a task-relative account of completion, explaining why some tasks admit full mechanization, others permit only asymptotic or approximate completion, and still others resist completion altogether.

While this perspective intersects with themes explored in dynamic semantics, game-theoretic reasoning, and cognitive pragmatics, it does not aim to subsume or replace these approaches. Its contribution is orthogonal: to clarify the structural conditions under which task completion is possible or impossible. A systematic comparison with adjacent frameworks, as well as with classical results on undecidability, lies beyond the scope of the present paper.

6.5. Consequences and Scope

The AI Boundary Thesis has several immediate consequences. A more detailed discussion follows in the next section.

First, it reframes certain apparent failures of artificial intelligence. When machines struggle with tasks such as natural language pragmatics, social coordination, or norm-governed interaction, the

source of difficulty may lie in the incompleteness of the task rather than in deficiencies of algorithm design.

Second, the thesis clarifies the role of approximation. Humans routinely perform logically-time incomplete tasks by accepting partial or contextually sufficient states. Machines designed to meet explicit completion criteria cannot do so without modifying the task specification itself.

7. Implications for Contemporary AI Systems and Philosophical Issues

The logical-time framework developed in this paper is not intended as a retrospective reinterpretation of existing AI techniques. Nevertheless, several prominent features of contemporary AI systems—particularly large language models—can be illuminated when viewed through the lens of logical time. This section briefly sketches these connections and outlines their philosophical and ethical implications.

7.1. No Claim About Human Superiority

The present thesis does not assert, imply, or presuppose that humans can complete tasks that machines cannot. On the contrary, it treats human and machine intelligence as sharing a common functional orientation: both are forms of intelligence insofar as they are evaluated by what they can do, achieve, or enable. Correspondingly, both may also be subject to structurally similar limitations. The focus of this paper is on the latter, on limits that arise from the structure of tasks themselves rather than from the nature of any particular agent.

Logical-time incompleteness is such a limit. It is a property of task specifications, not of human cognition or machine capability. Whether a task admits completion depends on the structure of its conceptual state space, admissible transitions, and completion conditions, not on whether the agent attempting it is biological or artificial.

But, human practices in dealing with logically incomplete tasks reveal several strategies that are instructive for artificial systems, some of which are already partially implemented in contemporary AI, and others of which remain underdeveloped.

First, humans often modify completion conditions, for example by tolerating vagueness, ambiguity, or underspecification, thereby transforming an incomplete task into one that is approximately or asymptotically completable. In the present framework, this corresponds to transforming a Class III task into a Class II task.

Second, humans may expand or restructure the conceptual state space itself—by introducing new concepts, representations, or theoretical frameworks—so that previously open-ended trajectories become closed. This amounts to altering the set of logical states LS so as to enable completion, effectively converting a Class III task into a Class II or even Class I task.

Third, and more fundamentally, humans are sometimes capable of recognizing incompleteness itself and responding not by approximation or reformulation, but by restraint. As Laozi observed, “*道可道，非常道*”; and as Wittgenstein concluded, “*Whereof one cannot speak, thereof one must be silent.*” (Laozi, Daodejing; Wittgenstein 1922) These do not overcome incompleteness; they recognize it.

This response does not demonstrate human superiority, but rather illustrates different modes of engaging with tasks whose logical-time structure precludes full completion. They underscore the central claim of this paper: that incompleteness is not a failure of intelligence, but a structural feature of certain tasks, and that recognizing this feature is itself an essential component of a mature theory of artificial intelligence.

7.2. Boundary Does Not Imply a Limit on AI Progress

The thesis advanced in this paper concerns the realizability of tasks **as formally specified**, rather than the long-term development or potential of machine cognition. A structural boundary on task completion does not entail a ceiling on empirical performance, nor does it preclude substantial progress through approximation, reformulation, or relaxation of completion conditions.

On the contrary, clarifying where structural boundaries arise can contribute positively to AI research by making explicit which aspects of a task resist full completion and why. Such clarification enables more informed allocation of conceptual and computational resources, guiding efforts toward architectures, representations, and evaluation criteria that are better aligned with the structural properties of the task.

In this sense, the present framework is intended not as a pessimistic diagnosis, but as an analytic tool: one that helps distinguish between limitations that stem from contingent technical constraints and those that arise from the intrinsic topology of the task itself.

7.3. *The Manifestation of Logical Time in Contemporary AI Technology*

Recent developments in artificial intelligence, particularly in large language models, exhibit structural patterns that closely resonate with the notion of logical time introduced in this paper. From the Transformer architecture to Universal Transformers and, more recently, chain-of-thought prompting, one can retrospectively identify mechanisms that implicitly allocate different amounts of computation to different conceptual stages of a task.

7.3.1. Logical Time and Attention Mechanisms

The idea of logical time appears, albeit implicitly, in existing architectures. A notable example is the transition from the standard Transformer model introduced in Attention Is All You Need to the Universal Transformer (Vaswani et al. 2017; Dehghani et al. 2018). As discussed in the latter work, the Universal Transformer introduces adaptive computation, allowing the model to perform more processing steps on elements that require greater contextual disambiguation.

Consider the example sentence “I arrived at the bank after crossing the river.” While tokens such as “I” and “river” are relatively unambiguous, the token “bank” requires additional semantic context to determine its meaning. In a standard Transformer, computational effort is distributed uniformly across tokens. In contrast, the Universal Transformer allows more computational steps to be allocated to ambiguous tokens like “bank,” while clearer tokens require fewer steps.

From the perspective of logical time, this can be interpreted as follows: certain conceptual elements require a longer logical-time trajectory to reach a locally completed interpretive state. Tasks that demand such extended logical-time progression naturally incur greater computational cost, and architectures that accommodate this asymmetry tend to exhibit improved performance.

7.3.2. The Physical Mapping of Logical Time: Chain-of-Thought

The emergence of chain-of-thought (CoT) prompting provides a particularly vivid illustration of how logical time can be externalized in contemporary models. A chain of thought consists of a sequence of intermediate natural-language reasoning steps leading to a final answer (Wei et al., 2022). Rather than compressing all reasoning into a single forward pass, CoT allows multi-step reasoning to unfold explicitly across a sequence of generated tokens.

From a structural standpoint, this can be understood as a partial physical realization of logical time. Reasoning steps that would otherwise be implicit or compressed are instead expanded across the model’s output sequence, allowing additional computational resources to be devoted to tasks that require longer logical-time trajectories. In this sense, chain-of-thought prompting trades physical resources—most notably context length—for extended logical-time progression. This serves as a concrete real-world illustration of the approach discussed earlier, namely, rendering a logically-time incompletable task logically-time completable by appropriately adjusting its completion conditions. From the present framework’s perspective, this can be understood as transforming certain Class III tasks into Class II tasks by relaxing completion conditions and permitting approximate intermediate states.

7.3.3. Context Windows as Physical Bounds on Logical Time

The finite context window of contemporary language models imposes a hard upper bound on the length of conceptual or inferential trajectories that can be processed within a single interaction. From the perspective of logical time, this bound can be understood as a physical cutoff on the unfolding of task-relative conceptual trajectories.

This helps explain why such models often perform well on tasks requiring moderate multi-step reasoning, yet fail abruptly once the depth of required conceptual unfolding exceeds the available context. These failures are commonly described in terms of memory limitations or scaling constraints. From the present perspective, however, they can also be understood as reflecting a structural mismatch: tasks whose completion conditions require unbounded logical-time unfolding are being forced into bounded physical realizations.

In this restricted but instructive sense, context limits provide a concrete illustration of logical-time incompleteness under physical constraints. Tasks whose logical-time structure lacks a terminating completion state will, when subjected to sufficiently extended reasoning, eventually encounter this physical ceiling. The resulting failures should therefore be understood not merely as empirical shortcomings of particular models, but as manifestations of a deeper structural tension between unbounded logical-time requirements and finite physical resources.

7.4. Approximate Completion and Large Language Models

Approximate completion occupies an important place in the present framework. This notion provides a key to understanding several persistent controversies and long-standing conceptual confusions in contemporary AI research.

Indeed, once the threefold classification of tasks is introduced in Section 5.2, a natural question arises. You claim that linguistic disambiguation is a task that is logically-time incomplete; yet in the real world, humans use language effectively, and in the machine domain, contemporary large language models perform remarkably well in language processing. Does this not indicate that an allegedly incomplete task has, in fact, already been completed?

A closer analysis reveals that what is achieved in practice is not genuine completion, but rather a form of **approximate—or pseudo—completion**. Human linguistic communication tolerates indeterminacy: ordinary language use does not eliminate all ambiguity, nor does it require fully determinate interpretations at every stage of interaction. Communication succeeds precisely because the task's completion conditions are relaxed or reformulated. In principle, tasks that are logically incomplete are treated as admitting satisfactory stopping points, thereby transforming Class III tasks into Class II tasks. It is precisely this intrinsic indeterminacy of natural language that has motivated the development of more rigorous scientific languages, such as mathematical and formal languages.

The empirical success of large language models follows the same pattern. It must be made explicit—and emphasized—that the effectiveness of LLMs does not stem from having resolved semantic ambiguity, but rather from a recharacterization of the task itself: the goal shifts from producing determinate interpretations to predicting the next token probabilistically. This reformulation converts a task that is fundamentally incomplete into one that is asymptotically convergent, thereby rendering it practically tractable.

Nevertheless, such success should be interpreted with caution. Probabilistic convergence is not equivalent to logical closure, and truncation is not completion. The convergence of a loss function to a (possibly local) minimum does not entail logical truth or conceptual completion. Current AI training objectives prioritize statistical proximity, which can obscure the structural fact that certain tasks remain logically-time incomplete. In this light, the 'hallucinations' observed in LLMs can be understood as the structural byproduct of forcibly truncating tasks that are, in terms of logical time, essentially incomplete.

Precisely because approximate completion replaces genuine completion, a residual error space is unavoidable. As a result, occasional failure, inconsistency, or hallucination is not merely an implementation flaw, but a structural consequence of asymptotic completion.

At the same time, the practical importance of approximate completion is undeniable. A central open question for future work is whether there exist principled methods for systematically transforming Class III tasks into Class II tasks.

7.5. Implications for AI Ethics : AI Risk and Pseudo-Completion

As noted at the outset of this paper, questions concerning the boundaries of artificial intelligence are no longer of purely theoretical interest; they increasingly bear on social trust, institutional decision-making, and ethical risk. The notion of logical-time limits offers a structural perspective from which certain contemporary concerns about AI systems can be more clearly diagnosed. Building on the analysis of approximate completion developed in Section 7.4, this section does not revisit the underlying structural arguments. Instead, it examines their ethical implications, focusing on how pseudo-completion and the resulting appearance of determinacy give rise to distinctive forms of AI risk.

Within the proposed classification of tasks, two closely related sources of ethical risk become particularly salient: pseudo-completion and the resulting pseudo-determinism. In earlier stages of AI development, such misattributions were relatively easy to avoid, as system performance was visibly limited. However, the impressive fluency and apparent competence of contemporary large models make such misjudgments increasingly likely.

Such systems, despite their apparent competence, may perform well within a tolerable margin of error, yet the very existence of this margin implies that error is structurally unavoidable.

From an engineering standpoint, partial completion may often be sufficient for practical use. Ethically, however, conflating “being sufficient for use” with “being fully determinate or reliable” is hazardous. In high-stakes domains, such as judicial decision support or autonomous ethical decision-making, treating outputs generated under logical-time incompleteness as fully determinate introduces a form of pseudo-determinism. The appearance of certainty masks the absence of a principled completion state, thereby inviting misplaced trust.

Recognizing this structural risk does not entail rejecting AI systems in such domains outright. Rather, it underscores the need for calibrated trust and for institutional safeguards that acknowledge the limits imposed by task structure itself.

Related concerns arise in discussions of alignment and value specification, though a detailed treatment lies beyond the scope of this paper.

8. Conclusion

This paper has proposed a task-relative framework for analyzing the limits of artificial intelligence through the notion of logical time. Rather than approaching AI boundaries in terms of computational power, learning capacity, or architectural sophistication, the framework shifts attention to the **structural conditions under which tasks admit completion at all**.

By characterizing tasks in terms of conceptual state spaces, admissible transitions, and completion conditions, logical time provides a dimension orthogonal to traditional measures of complexity or intelligence. Within this framework, **logical-time incompleteness** identifies tasks whose completion conditions are structurally inaccessible, regardless of the agent attempting the task or the resources available. Such limits arise not from insufficient realization, but from the topology of the task itself.

On this basis, the paper has introduced a threefold classification of tasks: those that are fully completable, those that admit only approximate or asymptotic completion, and those that are logically-time incomplete. This classification explains a range of familiar phenomena in contemporary AI: superhuman performance on formally well-structured tasks, persistent difficulty

with seemingly elementary human activities, the central role of approximation in successful systems, and the emergence of pseudo-completion and pseudo-determinism in high-stakes applications.

A central contribution of the paper lies in clarifying a category distinction that is often left implicit or conflated: the distinction between **machine realization** and **task completion**. Advances in realization—improved models, training regimes, or computational efficiency—do not by themselves alter the completion status of tasks whose logical-time structure blocks termination. Recognizing this distinction allows for a more precise understanding of both the achievements and the limitations of current AI systems.

The framework also sheds light on recent developments in large language models, including chain-of-thought prompting and context-window constraints, as strategies for managing logical time rather than overcoming its limits. Their success depends on transforming incompletable tasks into approximable ones, not on eliminating structural incompleteness itself.

Several limitations of the present work should be acknowledged. The framework is intentionally abstract and task-relative; it does not offer algorithmic criteria for identifying incompleteness in concrete systems, nor does it provide a full comparative analysis with adjacent approaches in semantics, game theory, or cognitive science. These omissions are deliberate. The aim has been to establish a conceptual boundary, not to subsume existing theories or propose new architectures.

Future work may extend this framework in multiple directions: by developing diagnostic tools for identifying task class in applied settings; by studying how agents—human or artificial—systematically reformulate tasks to shift their logical-time structure; or by integrating logical-time analysis with empirical studies of reasoning, interaction, and decision-making.

The broader implication is a call for conceptual discipline in discussions of artificial intelligence. Not all failures indicate insufficient intelligence, and not all successes amount to completion. Some limits persist even under idealized assumptions, because they are embedded in the structure of the tasks themselves. Logical time offers a way to make these limits explicit—and, in doing so, to better understand both what artificial intelligence can achieve and where its boundaries lie.

References

- Aczel, P. (1988). *Non-well-founded sets*. Stanford, CA: CSLI Publications.
- Davidson, D. (1973). Radical interpretation. *Dialectica*, 27(3–4), 314–328.
- Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2018). Universal Transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Gödel, K. (1931). On formally undecidable propositions of Principia Mathematica and related systems. *Monatshefte für Mathematik und Physik*, 38(1), 173–198.
- Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics, Vol. 3: Speech Acts* (pp. 41–58). New York: Academic Press.
- Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42(1–3), 335–346.
- Hayes, P. J., & Ford, K. M. (1995). Turing Test considered harmful. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 972–977.
- Kamp, H. (1981). A theory of truth and semantic representation. In J. A. G. Groenendijk, T. M. V. Janssen, & M. B. J. Stokhof (Eds.), *Formal methods in the study of language* (pp. 277–322). Amsterdam: Mathematical Centre Tracts.
- Laozi. (2001). *Daodejing* (P. J. Ivanhoe, Trans.). Indianapolis: Hackett Publishing. (Original work published c. 4th century BC).
- Moravec, H. (1988). *Mind children: The future of robot and human intelligence*. Cambridge, MA: Harvard University Press.
- Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3), 113–126.
- Quine, W. V. O. (1960). *Word and Object*. Cambridge, MA: MIT Press.
- Recanati, F. (2004). *Literal Meaning*. Cambridge: Cambridge University Press.

- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge: Cambridge University Press.
- Simon, H. A. (1957). *Models of man: Social and rational*. New York: Wiley.
- Sipser, M. (2013). *Introduction to the Theory of Computation* (3rd ed.). Cengage Learning.
- Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), 230–265.
- van Benthem, J. (1996). *Exploring Logical Dynamics*. Stanford, CA: CSLI Publications.
- Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.
- Wittgenstein, L. (1922). *Tractatus logico-philosophicus*. London: Routledge & Kegan Paul.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.