The Logical Impossibility of Artificial General Intelligence (AGI)

Anonymous authors
Paper under double-blind review

Abstract

This paper proves the logical impossibility of Artificial General Intelligence (AGI) through 8 proofs, with 7 independent proofs and the 8th proof being based on 4 independent proofs but with its independent importance. We give pointers to the research direction in Artificial Intelligence (AI) based on the logic of this paper proving AGI as logically impossible.

1 Introduction

The question of whether machines can think, first posed by Turing (1950), has been central to Artificial Intelligence (AI) research. This paper presents 8 proofs that Artificial General Intelligence (AGI) is logically impossible, with 7 independent proofs and the 8th proof being based on 4 independent proofs but with its independent importance.

2 Material and methods

This paper employs a comparative logical analysis methodology that:

1. Identifies core features of general intelligence as empirically demonstrated by human beings
2. Analyzes inherent limitations of programmed systems through formal logical proofs
3. Demonstrates logical impossibility of AGI through 8 proofs (7 independent proofs and the 8th proof being based on 4 independent proofs but with its independent importance) by showing the fundamental gaps between human general intelligence capabilities and programmed system limitations

The proofs utilize:

1. Experimental logic (for verifying human capabilities)
2. Formal logic (for analyzing programmed system limitations)
3. Logical necessity arguments

3 Definition of General Intelligence (GI)

3.1 Definition

Definition 1 (General Intelligence (GI)) *A system S exhibits general intelligence (GI) if and only if it possesses all of the following capabilities:*

1. ***Direct Understanding Capability (DUC):*** *The capability to:*
   - *Recognize truth without formal derivation*

- *Understand meaning without exhaustive symbolic processing*
- *Grasp self-evident principles without prior axioms*

2. ***Axiom Creation Capability (ACC):*** *The capability to create axioms A that are:*
   - *Not derivable from any existing set of axioms in the system*
   - *Self-evidently true without formal proof*
   - *Usable as foundations for new concepts, informal logical systems and formal systems*
3. ***Context-Independent Understanding Capability (CIUC):*** *The capability to comprehend and operate in arbitrary contexts C where:*
   - *C is not predefined within the system*
   - *No prior training or programming exists for C*
   - *Novel frameworks must be created to understand C*
4. ***Formal System Creation Capability (FSCC):*** *The capability to create new formal systems F where:*
   - *F consists of new symbols, axioms and rules of inference*
   - *F is not derivable from the existing formal systems*
   - *F can serve as a foundation for understanding novel phenomena*

Formally, for any system S:

$$GI(S) \iff DUC(S) \land ACC(S) \land CIUC(S) \land FSCC(S)$$

Where:

- *$GI(S)$ denotes that system S has general intelligence*
- *$DUC(S)$ denotes that system S has direct understanding capability*
- *$ACC(S)$ denotes that system S has axiom creation capability*
- *$CIUC(S)$ denotes that system S has context-independent understanding capability*
- *$FSCC(S)$ denotes that system S has formal system creation capability*

3.2 Justification of the Definition

The definition of general intelligence through these four specific capabilities ($ACC, CIUC, FSCC, DUC$) is both necessary and sufficient because:

1. **Completeness:** These capabilities collectively cover all essential aspects of general intelligence:
   - Direct understanding capability (DUC) encompasses the fundamental nature of intelligent comprehension
   - Axiom creation capability (ACC) captures the capability to establish new foundations of knowledge
   - Context-independent understanding capability ($CIUC$) addresses the capability to handle truly novel situations
   - Formal system creation capability ($FSCC$) represents the capability to create new formal systems for understanding anything new vis-a-vis the existing knowledge of the system and adding it to the existing knowledge of the system

2. **Independence:** While ACC is necessary for the other two capabilities, $CIUC$ and $FSCC$, each capability represents a distinct aspect of general intelligence:
   - DUC addresses immediate comprehension
   - ACC focuses on the foundational truth creation
   - $CIUC$ deals with adaptation to novelty
   - $FSCC$ concerns systematic knowledge organization
3. **Observability:** These capabilities are demonstrably present in human intelligence:
   - Human beings exhibit direct understanding of self-evident truths like the fire gives heat and light, water is wet, ice is cold, etc.
   - Human beings create new axioms in various fields, such as formal logic, mathematics, psychology, philosophy, etc.
   - Human beings adapt to completely novel situations
   - Human beings create new formal systems of knowledge
4. **Necessity:** The absence of any of these capabilities would result in a system that is:
   - Restricted to purely formal derivations
   - Limited to the existing knowledge frameworks
   - Unable to truly adapt to novelty
   - Incapable of creating new formal systems of understanding

Thus, all these 4 capabilities are necessary and sufficient for a system to have general intelligence.

3.3 Necessity of Axiom Creation Capability (ACC) Theorem

Theorem 2 (Necessity of Axiom Creation Capability (ACC)**)** *Axiom creation capability* (ACC) *is necessary for the other two capabilities,* $CIUC$ *and* $FSCC$*.*

Proof 1 *Consider each of the other two capabilities,* $CIUC$ *and* $FSCC$*:*

1. *Context-independent understanding capability* ($CIUC$) *requires creating new axioms for novel contexts as no pre-existing axioms can cover all possible contexts*
2. *Formal system creation capability* ($FSCC$) *necessarily involves creating new axioms as the foundation of any new formal system*

Therefore, ACC is necessary for the other two capabilities of CIUC and FSCC:

$$\forall S : (CIUC(S) \vee FSCC(S)) \implies ACC(S)$$

3.4 Independence of Direct Understanding Capability (DUC)

Direct understanding capability (DUC) is an independent capability among all 4 general intelligence capabilities as explained below:

1. Direct understanding can occur without creating new axioms - we can immediately grasp self-evident principles using the existing mental frameworks
2. DUC involves the recognition of truth and meaning, without derivation, that can happen without establishing new self-evident foundational principles or axioms
3. The ability to “understand meaning without exhaustive symbolic processing" represents a distinct cognitive capability separate from axiom creation

3.5 Centrality of Axiom Creation Capability (ACC) Theorem

Theorem 3 (Centrality of Axiom Creation Capability (ACC)**)** *Axiom creation capability* (ACC) *is both necessary and central to general intelligence.*

Proof 2 *From the " 1. Definition of General Intelligence (GI)" and " 2. Necessity of Axiom Creation Capability* (ACC) *Theorem":*

1. $GI(S)$ *requires all four capabilities*
2. $ACC(S)$ *is necessary for the other two capabilities of* $CIUC(S)$ *and* $FSCC(S)$
3. *Therefore,* $ACC(S)$ *is both necessary and central for* $GI(S)$

Formally:

$$\forall S : GI(S) \implies ACC(S)$$

4 Definition of Artificial General Intelligence (AGI)

Definition 4 (Artificial General Intelligence (AGI)) *Artificial General Intelligence (AGI) is the capability of Artificial Intelligence or programmed systems to show General Intelligence (GI).*

5 Proof 1. Axiom Creation Limitation Proof

5.1 Definition of Axiom Creation (AC)

Definition 5 (Axiom Creation) *An axiom creation* (AC) *operation on a system* S *produces an axiom* a *such that:*

1. $a \notin D(S)$ *where* $D(S)$ *is the deductive closure of* S
2. a *is self-evidently true*
3. a *can serve as a foundation for new formal systems*

5.2 Formal System Limitation Theorem

Theorem 6 (Formal System Limitation) *No formal system* F *can contain within itself the means to create new axioms.*

Proof 3 *Suppose for contradiction that a formal system* F *could create a new axiom* a*. Then:*

1. *Since* F *is a formal system, it must operate according to defined rules* R *of inference applied to its existing symbols and axioms*
2. *Any output of* F*, including* a*, must be derived through the application of these rules to the existing axioms and symbols. Therefore:*
$$a \in D(F)$$
where $D(F)$ *is the deductive closure of* F*.*
3. *However, by the definition of axiom creation:*
$$a \notin D(F)$$
4. *This is a contradiction. The axiom* a *cannot simultaneously be:*

- *Within the deductive closure of F (as it was created by F's rules)*
- *Outside the deductive closure of F (as required by the definition of axiom creation)*

5. *Therefore, our assumption must be false and no formal system can create new axioms*

This proof shows that any "new" statement generated by a formal system must be derivable from the existing axioms and, thus, is not truly new. This fundamental limitation arises from the fact that formal systems can only transform and combine the existing knowledge according to their rules of inference, while axiom creation requires establishing new foundational truths that are not derivable from the existing knowledge.

5.3 Proof

Theorem 7 (The Logical Impossibility of Artificial General Intelligence (AGI)) *Artificial General Intelligence (AGI) is logically impossible.*

Proof 4 *The proof follows from three fundamental facts:*

1. *All programmed systems (whether digital, quantum or analog) are formal systems*
2. *All formal systems require axioms*
3. *Axioms cannot be created within formal systems as proved in " 6. Formal System Limitation Theorem" earlier, but their creation requires the general intelligence of human beings*

Therefore, programming cannot create axioms but can only create theorems and lemmas, which are the derived results of the already programmed rules of inference applied to the defined symbols and axioms of the programmed system, making AGI logically impossible.

5.4 Analysis of the Proof

5.4.1 Nature of Formal Systems

Formal systems, as explained by Gödel (1931), require:

- A finite set of symbols (the alphabet)
- A finite set of axioms, which are statements taken to be true to serve as the premises or starting points for further reasoning and arguments
- A finite set of rules of inference

5.4.2 Programming as Formal System

Programming is fundamentally a formal system because it:

- Uses defined symbols like keywords, operators, etc.
- Operates on defined axioms of arithmetic operations, control flow rules, etc.
- Follows defined rules like syntax rules, type systems, semantic rules, etc.

5.4.3 Axiom Creation

Axioms cannot be created within formal systems because:

- They form the foundation of formal systems that, by logical necessity, must exist prior to formal systems themselves and cannot be derived by formal systems themselves
- Their truth must be understood directly
- Their creation requires the general intelligence of human beings

5.5 Comprehensive Refutation of Counterarguments

5.5.1 The Evolution Argument

Counterargument: Programmed systems could evolve to create new axioms, just as biological evolution produced intelligence.

Refutation:

- Evolution in programming is itself a formal system operation
- Every evolutionary algorithm requires defined rules for mutation, selection and reproduction
- All these rules are based on the programmed axioms
- Cannot transcend the original axiomatic framework
- Any “new" development is still the combination of the existing axioms
- Biological evolution comparison is invalid because it is an experimentally proven fact of our own human existence that biological systems are not formal systems as they create axioms and various formal systems based on various symbols, axioms and rules, and have direct understanding and context-free understanding capabilities

5.5.2 The Learning Argument

Counterargument: Advanced machine learning could develop the capability to create axioms through experience.

Refutation:

- All machine learning algorithms are formal systems because they require defined learning rules, use specified optimization criteria and follow the programmed frameworks
- In formal systems, learning is pattern recognition within the given axiom space
- Cannot transcend foundational axioms because learning rules, pattern recognition and optimization are based on axioms
- “Learning" is applying the existing axioms, not creating new ones
- No amount of learning can transcend axiomatic foundation

5.5.3 The Emergence Argument

Counterargument: Sufficient complexity of programmed systems could lead to the emergence of axiom creation capability.

Refutation:

- The emergence in programmed systems is still bound by defined axioms and defined rules
- The complexity of programmed systems does not change their fundamental nature because complex combinations are still combinations and do not offer any escape from formal system nature and, thus, cannot transcend axiom foundation
- Emergence cannot create truly new axioms because all emergence is based on the existing rules that themselves get applied to the existing axioms, making any transcendence impossible

5.5.4 The Quantum Computing Argument

Counterargument: Quantum computation's fundamentally different nature could allow axiom creation.

Refutation:

- Quantum computers, despite using quantum mechanical principles, are still programmed systems that use defined quantum axioms, follow quantum rules and operate within a formal framework
- The quantum nature of computation adds computational capabilities based on quantum mechanics and changes how computations happen but does not change the need for axioms, formal system nature and fundamental limitations
- Thus, quantum uncertainty cannot enable axiom creation through the transcendence of the formal system nature of quantum computers

5.5.5 The Simulation Argument

Counterargument: The perfect simulation of general intelligence would include axiom creation capability.

Refutation:

- A simulation, no matter how advanced, is still a programmed system that is a formal system based on the programmed axioms and, thus, cannot transcend its programmed foundation
- Even a "perfect" simulation must still follow its programmed rules and, thus, can only use its existing axioms and cannot create truly new axioms
- Simulation paradox: to simulate axiom creation, the programmed system would need axioms about axiom creation leading to an infinite regress

5.5.6 The Awareness Emergence Argument

Counterargument: Awareness could emerge in a programmed system enabling axiom creation.

Refutation:

- Awareness, as we directly know, requires non-formal axiom creation, direct understanding, context-free understanding and formal system creation capabilities
- Formal systems cannot create awareness as they cannot, by logical necessity, transcend their formal nature bound by their defined axioms and cannot generate axiom creation, direct understanding, context-free understanding and formal system creation capabilities

5.5.7 The Self-Modification Argument

Counterargument: Programmed systems could modify their own axioms.

Refutation:

- Self-modification still requires axioms and rules for modification
- Cannot escape original axioms due to the programmed system having a formal nature
- Modification $\neq$ creation because modifications are within the programmed system based on the existing axioms with no possibility of transcendence

5.5.8 The Human Brain Argument

Counterargument: Since human brain is a physical system, a sufficiently complex physical computing system could achieve general intelligence.

Refutation:

- Human brain, even if considered just a physical system, is qualitatively different than programmed systems as proven by our direct experience that shows it has awareness, direct understanding, context-independent understanding and the capability to create axioms and formal systems based on them
- Programmed systems remain formal systems regardless of physical complexity, processing power and architecture type

5.5.9 The Unknown Mechanism Argument

Counterargument: We do not know how humans create axioms, so programmed systems might have an unknown way.

Refutation:

- Not knowing how humans create axioms does not change the logical limitations of programmed systems, making it impossible for them to create axioms
- Any programmed system must still be a formal system programmed using axioms created by human beings, based on defined rules and limited by its formal system nature
- “Unknown" does not mean “possible" when something is proven logically impossible and limited by logical necessity

5.5.10 The Different Type of General Intelligence Argument

Counterargument: Programmed systems could develop a different but equivalent form of general intelligence not needing axiom creation.

Refutation:

- General intelligence, by logical necessity, requires axiom creation, direct understanding, context-independent understanding and formal system creation capabilities
- No equivalent form of general intelligence is possible because this is a fundamental limitation of formal systems that they cannot escape axiom dependence
- Any machine intelligence must remain based on formal systems and, hence, dependent on given axioms and, thus, fundamentally incapable of axiom creation and, hence, general intelligence

5.5.11 The Hybrid System Argument

Counterargument: Combining multiple approaches could overcome individual limitations and enable axiom creation capability.

Refutation:

- The combination of formal systems is still a formal system and, thus, axiom-dependent and fundamentally limited
- No emergent transcendence through any combination is possible because all components are formal systems
- Thus, hybrid nature does not provide axiom creation capability and, hence, general intelligence

5.5.12 The Gradual Development Argument

Counterargument: Programmed systems could gradually develop axiom creation through incremental advances.

Refutation:

- Any gradual advance in programmed systems is still within a formal system using the existing axioms and under fundamental limitations
- No amount of time or gradual change can create new axioms and transcend the formal system nature
- All development remains bound by the original formal system nature and dependence on the existing axioms

5.5.13 The Human Augmentation Argument

Counterargument: Programmed systems could leverage human axiom creation capability.

Refutation:

- This is human beings helping programmed systems with actual axiom creation
- Programmed systems still remain formal systems with dependence on the existing axioms and are unable to create axioms themselves
- Proves the main point that programmed systems need human intelligence for axiom creation
- They cannot achieve axiom creation capability on their own
- Leveraging human axiom creation capability is not AGI but is human-dependent AI

5.5.14 The Mathematical Platonism Argument

Counterargument: If mathematical truths exist independently, programmed systems could discover axioms rather than create them.

Refutation:

- Even the discovery of mathematical truths requires direct understanding to recognize the truth through non-formal insight to grasp self-evidence and axiom creation capability to establish new foundations
- Programmed systems can only follow programmed rules to process the existing axioms and can work only within formal system limitations
- Mathematical Platonism does not change the need for direct understanding, the formal system limitations and the logical impossibility of programmed systems discovering new axioms

5.5.15 The Self-Reference Argument

Counterargument: Since formal systems can refer to themselves (as in Gödel's proof of the First Incompleteness Theorem (Gödel, 1931)), they might transcend to create axioms.

Refutation:

- Self-reference in formal systems still uses the existing axioms and rules
- Self-reference does not create new axioms but operates within the given axioms, uses the existing rules and actually demonstrates formal system limitations
- This actually proves our point that programmed systems cannot transcend their axioms even while referring to themselves

5.6 Completeness of Counterarguments Analysis

This section demonstrates that our analysis of counterarguments is complete, covering all logically possible challenges to the axiom creation limitation proof of AGI impossibility.

5.6.1 Categories of Counterarguments

The counterarguments fall into the fundamental categories that exhaust all possible challenges:

1. **Physical Approaches:**
   - The Quantum Computing Argument
   - The Human Brain Argument
   - All physical implementations must be formal systems
2. **Computational Approaches:**
   - The Evolution Argument
   - The Learning Argument
   - The Simulation Argument
   - The Hybrid System Argument
   - All computational approaches are formal systems by definition
3. **Process Approaches:**
   - The Self-Modification Argument
   - The Gradual Development Argument
   - The Human Augmentation Argument
   - All processes in programmed systems are formal system operations
4. **Emergent Approaches:**
   - The Emergence Argument
   - The Awareness Emergence Argument
   - All emergence in programmed systems is bound by formal system nature
5. **Meta Approaches:**
   - The Self-Reference Argument
   - All meta-level operations are still formal system operations
6. **Philosophical Approaches:**
   - The Unknown Mechanism Argument
   - The Different Type of General Intelligence Argument
   - The Mathematical Platonism Argument
   - All conceptual frameworks still require formal implementation

5.6.2 Completeness of Coverage

This categorization is complete because:

1. **Implementation Exhaustion:** Any proposed AGI implementation must either:
   - Use known mechanisms (covered by specific arguments)
   - Use unknown mechanisms (covered by the Unknown Mechanism Argument)
   - Use combinations (covered by the Hybrid System Argument)

2. **Logical Exhaustion:** Any challenge to the proof must either:
   - Question formal system nature (covered by Physical Approaches as mentioned earlier)
   - Propose alternative mechanisms (covered by various arguments)
   - Suggest transcendence of limitations (covered by Meta Approaches as mentioned earlier)
3. **Conceptual Exhaustion:** Any conceptual basis must either:
   - Work within formal systems (covered by main proof)
   - Attempt to transcend formal systems (covered by various arguments)
   - Propose alternative frameworks (covered by " 5.5.10. The Different Type of General Intelligence Argument")

5.6.3 Fundamental Limitation

All counterarguments ultimately face the same insurmountable obstacle:

- Any programmed system must be a formal system
- Formal systems cannot create new axioms
- Without axiom creation, general intelligence is impossible
- Therefore, AGI is logically impossible

5.6.4 Conclusion

The analysis of counterarguments is complete because:

- All possible categories of implementation are covered
- All possible types of challenges are addressed
- All face the same fundamental limitation
- No other categories are logically possible

This completeness, combined with the rigor of individual refutations, strengthens the main proof by showing that no possible avenue exists for achieving AGI.

6 Proof 2. Logical Potential Limitation Proof

This section presents a fundamental characteristic of human intelligence that further demonstrates the impossibility of AGI.

6.1 Shared Logical Potential

A key observation about axioms reveals another unbridgeable gap between human intelligence and formal systems:

1. Logical potential is the potential to create any new understanding that cannot be derived from one's existing knowledge by a human being irrespective of whether it has been discovered by others or not
2. Axioms exist as logical potential in human minds prior to their formalization
3. This logical potential is shared across all human minds
4. When one human mind actualizes this potential into explicit axioms, other human minds can directly understand and accept them, and they become part of human knowledge
5. This universal recognition demonstrates that axioms are not arbitrary creations but are the actualizations of shared logical potential

6.2 Evidence of Shared Logical Potential

Several phenomena demonstrate this shared logical potential:

6.2.1 Universal Acceptance

- Once properly formalized, logical and mathematical axioms achieve immediate recognition across different minds
- This acceptance crosses cultural, spatial and temporal boundaries
- The recognition is based on direct understanding, not formal proof

6.2.2 Cross-Cultural Development

- Different cultures independently developed similar logical foundations
- Basic logical and mathematical truths were recognized across civilizations
- This convergence demonstrates access to the same logical potential in all human beings

6.2.3 Axiom Validation

- Human minds can recognize valid versus invalid axiom systems
- This recognition occurs through direct understanding
- No formal proof is needed for basic axiom acceptance

6.3 Implications for AGI Impossibility

This understanding of axioms' nature strengthens the AGI impossibility proof:

6.3.1 Fundamental Human Capability

- Human minds possess shared logical potential
- Human minds can recognize this logical potential
- Human minds can actualize this logical potential into explicit axioms
- This enables the creation of new concepts, informal logical systems and formal systems

6.3.2 Formal System Limitations

- Formal systems have no access to logical potential
- They cannot recognize logical potential
- They cannot actualize logical potential into axioms
- They can only operate within the given axioms

6.3.3 Unbridgeable Gap

- The gap is not just in axiom creation
- It extends to the very foundation of logical understanding
- No formal system can replicate this capability
- This limitation is fundamental and categorical

6.4 Additional Support for the Axiom Creation Limitation Proof

This understanding provides additional support for our axiom creation limitation proof:

6.4.1 Explains the Foundation of General Intelligence

- Shows why human-created axioms are reliable
- Demonstrates why formal systems depend on human intelligence
- Reveals why axiom creation must precede concept formation, informal logical systems and formal systems

6.4.2 Demonstrates Necessity

- Access to logical potential is necessary for general intelligence
- Ability to actualize this logical potential is necessary for axiom creation
- Both capabilities are fundamentally beyond formal systems including programmed systems

6.4.3 Confirms Impossibility

- Proves that the limitation is not just practical but fundamental
- Shows that the gap cannot be bridged by any formal means
- Provides another independent proof of AGI impossibility

6.5 Relation to Other Arguments

This understanding illuminates and strengthens all our previous arguments, some of which are given below:

6.5.1 The Evolution Argument

- Explains why the evolution of formal systems cannot create new axioms
- Shows why complexity cannot bridge this fundamental gap
- Demonstrates why emergence cannot replicate this capability

6.5.2 The Learning Argument

- Reveals why machine learning cannot access logical potential
- Shows why pattern recognition differs from axiom understanding
- Explains why learning cannot develop axiom creation capability

6.5.3 The Simulation Argument

- Demonstrates why simulation cannot replicate logical potential
- Shows why even perfect simulation still lacks the fundamental capability of logical potential

This additional understanding of axioms' pre-existing nature as logical potential in human minds before they get created provides another fundamental and independent proof of AGI's logical impossibility, further strengthening our conclusion.

7 Proof 3. Understanding the Meaning Limitation Proof

7.1 Proof

Theorem 8 *Programmed systems cannot achieve general intelligence.*

Proof 5 (Proof by Contradiction) *1. Through experimental logic, we verify that general intelligence includes the capability to:*

- *Understand the meanings of various concepts independent of their representations and what it means to be an instance of a concept*
- *Understand the same meaning of a concept or its instance through its different representations*
- *Create new representations for the same meaning of a concept or its instance*
- *Comprehend the meaning of a concept or its instance beyond any specific representation*

2. This understanding capability extends to:

- *Understanding the meanings of the particular instances of a concept versus the meaning of the concept itself (for example, understanding this specific apple versus the concept of apple)*
- *Understanding the meanings of combinations (like understanding "red apple" combines the concepts of "red" and "apple")*
- *Understanding the relationships between various meanings (like how "fruit" relates to "apple")*
- *Understanding direct meanings (like "tree" means a plant with a trunk (s), branches and leaves)*
- *Understanding indirect or implied meanings (like "family tree" implies relationships rather than an actual tree)*
- *Understanding meanings in different contexts (like how "cool" means different things in different situations)*

3. These capabilities are crucial for all 4 capabilities of General Intelligence: Direct Understanding Capability (DUC), Axiom Creation Capability (ACC), Context-Independent Understanding Capability (CIUC) and Formal System Creation Capability (FSCC), as defined in " 1. Definition of General Intelligence (GI)"

4. Assume a programmed system can understand these meanings

5. By logical necessity:

- *Programmed systems can only process symbolic representations through binary or other numerical encoding (for example, converting "apple" to binary code)*
- *An concept or its instance can have infinite possible representations (like "apple1", "apple2", "apple3", ... continuing infinitely as natural numbers continue infinitely)*
- *No representation equals the meaning it represents (binary code for apple is not the understanding of what an apple is)*
- *If one claims that representation is the same as meaning, it is claiming that getting love from someone is the same as getting the word "love" written on a paper from him/her*
- *No combination of representations can bridge the gap to meaning*

6. This creates a fundamental contradiction:

- *Programmed systems must use some representation to process meaning*
- *But meaning exists independent of all representations*
- *This gap cannot be bridged through any computational means*
- *Makes understanding meaning logically impossible for programmed systems*

7. This impossibility applies to:

- *All types of meanings (whether abstract like "justice" or concrete like "chair")*

- *All levels of meanings (from simple direct meanings to complex metaphorical meanings)*
- *All combinations of meanings (from basic concepts to complex combinations)*
- *All relationships between meanings (how different concepts and their instances relate and interact)*

8. *Therefore, programmed systems cannot achieve general intelligence.*

7.2 Proof of Infinite Possible Representations

Consider the concept of “apple" fruit:

- One can create its infinite representations by:
  - Adding natural numbers sequentially to “apple" word: “apple1", “apple2", “apple3", ...
  - This process can continue infinitely as natural numbers continue infinitely
  - Each representation refers to the same meaning
  - Yet no representation equals the meaning itself
  - Applicable to all concepts and their instances through similar process and arguments
- This demonstrates:
  - Infinite possible representations exist for any concept or its instance
  - Each representation is still at the symbolic level
  - The meaning of a concept or its instance transcends all its representations
  - Programmed systems trapped at the representation level

7.3 Addressing Counter-Arguments

7.3.1 Counter-Argument 1: “Processing Multiple Representations of a Concept or Its Instance Enables Its Understanding"

- Claim: Processing multiple representations of a concept or its instance could lead to its understanding
- Refutation:
  - Multiple representations of a concept or its instance still remain at the representational level (each form, whether word, picture or description, is still just a representation)
  - No amount of representation processing can create the understanding of the meaning of a concept or its instance (having a million ways to represent “tree" does not create the understanding of what a tree actually is)
  - The quantity of representations of a concept or its instance cannot bridge the qualitative gap to its meaning that is independent of all its representations
  - Fundamental logical barrier remains

7.3.2 Counter-Argument 2: “Neural Networks or Other Pattern Matching Algorithms Learn the Meanings of Various Concepts and Their Instances"

- Claim: Neural networks or other pattern matching algorithms can extract the meanings of various concepts and their instances through pattern recognition
- Refutation:
  - Neural networks or other pattern matching algorithms still processing patterns of representations of various concepts and their instances (just matching patterns in pixels or text, not understanding what they mean)

- No mechanism to transcend the representational level (analyzing a billion tree images still does not create the understanding of “tree")
- Pattern matching $\neq$ understanding the meaning of a concept or its instance (recognizing tree patterns $\neq$ understanding what a tree is)
- Pattern matching remains within formal symbol manipulation as just more sophisticated computational operations

7.3.3 Counter-Argument 3: “Sophisticated Encoding Schemes Could Capture the Meanings of Various Concepts and Their Instances"

- Claim: Sophisticated encoding schemes could capture the meanings of various concepts and their instances
- Refutation:
  - Any encoding of a concept or its instance, howsoever sophisticated, is still a representation (even the most complex encoding of “tree" is still just another representation)
  - The meaning of a concept or its instance exists independent of all possible encodings (what a tree is exists independent of how we encode it)
  - No encoding can equal the meaning of a concept or its instance (the map is not the territory)
  - Logical impossibility remains

7.4 Understanding and Thinking Limitation Theorem

Theorem 9 (Understanding and Thinking Limitation) *Programmed systems cannot understand and think.*

Proof 6 *Consider the following:*

1. *Thinking requires producing thoughts involving various concepts, their instances and the combinations of these two*
2. *Understanding and thinking require understanding the meanings of various concepts, their instances and the combinations of these two*
3. *Programmed systems cannot understand the meanings of various concepts, their instances and the combination of these two as proved in “ 7. Understanding the Meaning Limitation Proof"*
4. *Therefore, programmed systems cannot understand and think*

8 Proof 4. Infinity Limitation Proof

Theorem 10 *Programmed systems cannot achieve general intelligence.*

Proof 7 *1. The capabilities required for general intelligence as defined in “ 1. Definition of General Intelligence (GI)" necessarily involve infinity in the following ways:*

- *Direct Understanding Capability (DUC) requires the ability to understand concepts that transcend finite representation, of which infinity is the most fundamental concept*
- *Axiom Creation Capability (ACC) requires creating axioms about unbounded domains, which necessarily involves understanding infinite possibilities*
- *Context-Independent Understanding Capability (CIUC) requires operating in arbitrary contexts, which form an infinite set of possible scenarios*
- *Formal System Creation Capability (FSCC) requires creating systems that can represent and operate on infinity, as demonstrated by human-created systems like calculus and set theory*

2. *A programmed system being an implementation of formal logic using a finite number of symbols, axioms and rules of inference is stuck at a meta-level problem:*
   - *Cannot truly conceptualize what infinity or infinite cardinality means beyond its predefined formal system*
   - *Can only work with specific predefined notations or symbols for infinity*
   - *Has no way to understand or define new types of infinite quantities or infinity of any new type that transcend its formal system*
3. *This limitation applies universally to:*
   - *Infinite sets, known or unknown*
   - *Infinite sequences of operations of any kind*
   - *Any new type of infinity we might discover/create in mathematics, physics, philosophy and any other branch of human knowledge*
4. *The history of human mathematical, scientific and philosophical thought demonstrates that understanding infinity is not optional but essential for general intelligence:*
   - *The development of natural numbers and operations on natural numbers like addition, subtraction, multiplication and division required conceptualizing infinity - same for other infinite cardinality sets, infinite sequences of mathematical operations and calculus through infinite division or limit*
   - *Physics and cosmology require understanding infinity to model the universe*
5. *Therefore, since programmed systems cannot truly understand infinity — which is essential for all four capabilities required for general intelligence — they cannot achieve general intelligence*
6. *Thus, this is an independent proof of the logical impossibility of Artificial General Intelligence (AGI)*

9 Proof 5. Continuity Limitation Proof

Theorem 11 *Programmed systems cannot achieve general intelligence.*

Proof 8 *1. The capabilities required for general intelligence as defined in " 1. Definition of General Intelligence (GI)" necessarily involve true continuity in the following ways:*

- ***Direct Understanding Capability (DUC)*** *requires the ability to understand continuous phenomena without reduction to discrete approximations, which includes:*
  - *Understanding the continuum of real numbers as a complete ordered field*
  - *Grasping the continuous flow of time and present moment*
  - *Comprehending continuous transformations as unified wholes*
  - *Apprehending continuous experiences without discretization*
- ***Axiom Creation Capability (ACC)*** *requires creating axioms about continuous domains, which necessarily involves:*
  - *Formulating axioms about continuous mathematical structures*
  - *Establishing foundational principles for continuous physical phenomena*
  - *Creating self-evident truths about continuous experiential states*
  - *Developing axioms about the relationship between discrete and continuous domains*
- ***Context-Independent Understanding Capability (CIUC)*** *requires operating in continuous contexts, which includes:*
  - *Understanding purely continuous domains (e.g., classical spacetime)*
  - *Operating in hybrid discrete-continuous contexts*
  - *Generating appropriate understanding for novel continuous phenomena*
  - *Moving between discrete and continuous representations without loss of understanding*

- ***Formal System Creation Capability (FSCC)*** *requires creating formal systems that capture true continuity, demonstrated by:*
  - *Human-created systems like calculus, topology and continuum mechanics*
  - *Mathematical structures that represent true continuity rather than approximations*
  - *Formal systems unifying discrete and continuous aspects of phenomena*
  - *Novel formal frameworks for previously unconceptualized continuous domains*

2. *A programmed system, being fundamentally discrete, faces insurmountable limitations regarding continuity:*
   - *All computation in programmed systems reduces to discrete operations on discrete data*
   - *Even when representing continuous mathematical objects:*
     - *Real numbers are represented through finite approximations*
     - *Continuous functions are discretized for computation*
     - *Differential equations are solved through discrete numerical methods*
     - *Limits are calculated through discrete approximation sequences*
   - *Any representation of continuity is necessarily:*
     - *Finite in precision*
     - *Discrete in nature*
     - *An approximation of true continuity*
     - *Unable to capture the fundamental nature of continuity itself*
   - *This limitation applies even to analog computers, which, despite measuring continuous physical properties, cannot achieve infinite precision in measurement and, thus, cannot truly capture continuity*

3. *The history of human mathematical and scientific thought demonstrates that understanding true continuity is essential for general intelligence:*
   - *The development of the concept of infinity and infinite cardinality and resulting mathematical concepts required the concept of pure continuity without any boundary*
   - *Calculus required a genuine understanding of continuity and limits*
   - *Physics requires continuous mathematical models for fundamental theories*
   - *Philosophy of mind involves continuous aspects of conscious experience*
   - *Human understanding of motion and change relies on genuinely continuous concepts*

4. *Therefore, since programmed systems cannot truly understand continuity — which is essential for all four capabilities required for general intelligence — they cannot achieve general intelligence*

5. *Thus, this is an independent proof of the logical impossibility of Artificial General Intelligence (AGI)*

10 Proof 6. Informal Logic Limitation Proof

Theorem 12 *Programmed systems cannot achieve general intelligence.*

Proof 9 *1. Through experimental logic, we observe that informal logic exists in human beings through our capability to:*

- *Create new words and meanings as needed*
- *Modify the existing language usage based on context*
- *Develop entirely new languages*
- *Understand meaning beyond literal definitions*
- *Adapt language rules organically without formal processes*
- *Create new axioms and new concepts and new formal systems based on new axioms*

- *Modify the existing formal systems based on new observations*
- *Generate novel logical frameworks in natural languages*

2. *The capabilities required for general intelligence as defined in " 1. Definition of General Intelligence (GI)" necessarily involve informal logical capability in the following ways:*
   - *Direct Understanding Capability (DUC) requires informal logical capability to have direct understanding of truth without formal derivation*
   - *Axiom Creation Capability (ACC) requires developing understanding that cannot be derived from any existing formal system and, hence, needs informal logical capability*
   - *Context-Independent Understanding Capability (CIUC) requires informal logical capability to understand arbitrary undefined contexts as formal logic operates only in already defined contexts*
   - *Formal System Creation Capability (FSCC) requires informal logical capability because in any formal system based on a fixed number of concepts, to avoid circular definition fallacy, at least one concept must be defined informally through informal logical capability*
3. *By logical necessity, any programmed system is bound to formal logic*
4. *Therefore, programmed systems cannot have informal logical capability*
5. *Therefore, programmed systems cannot achieve general intelligence*

11 Proof 7. Formal System Analysis Limitation Proof

Theorem 13 *Programmed systems cannot achieve general intelligence.*

Proof 10 *1. Through experimental logic, general intelligence requires the complete capability to analyze formal systems, including:*

- *Their symbols*
- *Their axioms*
- *Their rules of inference*

2. *The capabilities required for general intelligence as defined in " 1. Definition of General Intelligence (GI)" necessarily involve formal system analysis in the following ways:*
   - ***Direct Understanding Capability (DUC)** requires the ability to:*
     - *Directly understand the foundational elements of any formal system such as symbols, axioms and rules of inference as each of them is an atomic truth which cannot be understood in that formal system and must be understood on its own requiring direct understanding*
     - *Grasp the implicit meaning and limitations of symbols without recursive reference*
     - *Comprehend the nature and consequences of axioms outside their own framework*
     - *Recognize truth in formal statements without exhaustive symbolic processing*
   - ***Axiom Creation Capability (ACC)** requires creating axioms that transcend the existing formal systems, which necessarily involves:*
     - *Evaluating the logical powers of the existing formal systems*
     - *Creating new axioms that address limitations in the existing formal systems*
     - *Establishing new axioms that cannot be derived from within the existing formal systems*
     - *Developing meta-axioms about the relationships between different formal systems*
   - ***Context-Independent Understanding Capability (CIUC)** requires operating across formal systems, which includes:*
     - *Understanding the boundaries and limitations of any given formal system*
     - *Operating in contexts where formal systems themselves must be evaluated*
     - *Generating appropriate meta-frameworks for analyzing novel formal systems*
     - *Moving between different formal systems without being constrained by any single one*

- ***Formal System Creation Capability (FSCC)*** *requires creating new formal systems, which necessitates:*
  - *Complete analysis of the existing formal systems to identify their limitations*
  - *Creation of new symbols, axioms and rules of inference for new formal systems not derivable from the existing formal systems*
  - *Developing meta-formal systems for analyzing other formal systems*

3. *A programmed system operates entirely within its own formal system*

4. *A formal system operating within a formal system cannot analyze that same formal system due to the following reasons:*

   - *To analyze its own axioms, it needs meta-axioms, which are, by definition, not its part as whatever is defined by it becomes its axioms*
   - *Any formal system that attempts to fully analyze itself must use its own rules to evaluate those same rules, creating an irresolvable circular dependency*
   - *This limitation is demonstrated in fundamental results from mathematical logic - notably Gödel's Incompleteness Theorems (Gödel, 1931), Tarski's Undefinability Theorem (Tarski, 1983), and Kleene's Recursion Theorems (Kleene, 1938) - which collectively establish the inherent limitations of formal systems to analyze themselves.*

5. *This is not a partial limitation but a complete impossibility due to the programmed system being a formal system*

6. *Therefore, no programmed system can analyze its own formal logical structure*

7. *Therefore, programmed systems cannot achieve general intelligence*

12 NCI-CCI Framework for Universal Knowledge Creation and "Proof 8. NCI-CCI Limitation Proof"

12.1 NCI-CCI Framework for Universal Knowledge Creation

NCI-CCI framework of general intelligence consists of the following two capabilities:

1. **New Concept Intuition (NCI)**

Definition 14 (NCI) *The capability to create a concept that cannot be created through any combination of various concepts already present in the knowledge system.*

2. **Combinatorial Concept Intuition (CCI)**

Definition 15 (CCI) *The capability to create a concept through some combination of various concepts already present in the knowledge system.*

Theorem 16 (NCI-CCI Framework as the Only Possible Universal Theory of Knowledge Creation) *NCI-CCI framework is the only possible universal theory of knowledge creation.*

Proof 11 *1. By the law of excluded middle, for any new knowledge:*

- *Either it is derivable from the existing knowledge (CCI)*
- *Or it is not derivable from the existing knowledge (NCI)*
- *No third possibility can exist*

2. This categorization:

- *Is exhaustive (covers all possibilities)*
- *Is mutually exclusive (no overlap)*

- *Applies to all types of knowledge creation*

3. *Applies across all knowledge domains:*
   - *NCI and CCI capabilities independent of empirical experiences exist in fields like pure mathematics and logic*
   - *NCI and CCI capabilities dependent on empirical experiences exist in fields like physics, chemistry, biology and social sciences*
   - *This universality demonstrates the framework's comprehensive nature*
4. *Therefore:*
   - *NCI-CCI framework must be complete*
   - *No other framework is possible*
   - *It is the only possible universal theory of knowledge creation*

12.2 Dependence of NCI-CCI on Logical Potential and Understanding the Meaning of the Existing as Well as New Concepts

Only that concept can be created that is potentially already present as logical potential because the very nature of the universality of concepts requires it as explained in " 6. Logical Potential Limitation Proof".

Also, NCI and CCI require understanding the meanings of the existing concepts to recognize conceptual gaps in order to create new concepts and the capability to understand and also to validate whether a newly created concept is genuinely new. So, NCI and CCI are dependent on both logical potential and understanding the meanings of the existing as well as new concepts.

Irrespective of these prerequisites, NCI and CCI are independent capabilities of general intelligence because they are fundamentally new effects, even if arising due to the background of logical potential and understanding the meaning of the existing as well as new concepts present in general intelligence.

12.3 Proof 8. NCI-CCI Limitation Proof

Theorem 17 (NCI-CCI Limitation) *Programmed systems cannot achieve general intelligence.*

Proof 12 *1. Programmed systems, being formal systems:*

- *Are limited to manipulating the existing concepts at just symbolic or representational level*
- *Cannot transcend their programmed framework*

2. *They cannot implement New Concept Intuition (NCI) as well as Combinatorial Concept Intuition (CCI) because:*
   - *All outputs must be derivable from the existing concepts as already proved in the case of formal logic through " 5. Axiom Creation Limitation Proof" and in the case of informal logic through both this proof and " 10. Informal Logic Limitation Proof"*
   - *They cannot step outside their conceptual framework as:*
     - *New combinations still rely on the existing concepts and, for programmed systems, any supposed combination of the existing concepts are just symbol manipulations without any understanding*
     - *Only that new concept can be created that is potentially already present as logical potential because the very nature of the universality of concepts requires it as explained in " 6. Logical Potential Limitation Proof".*
   - *Also, NCI and CCI require understanding the meanings of the existing concepts to recognize conceptual gaps in order to create new concepts and the capability to understand and also to validate whether a newly created concept is genuinely new. So, NCI and CCI are dependent on both logical potential and understanding the meanings of the existing as well as new concepts.*

Thus, NCI and CCI, despite being fundamentally independent capabilities of general intelligence, are dependent on both logical potential and understanding the meanings of the existing as well as new concepts, none of which can be present in any programmed system as already proved in " 6. Logical Potential Limitation Proof" and " 7. Understanding the Meaning Limitation Proof".

3. *Therefore:*
   - *Programmed systems cannot achieve general intelligence*
   - *This limitation is fundamental and categorical*

12.4 Reliability Limitation of the Programmed Systems

Theorem 18 (Reliability Limitation of the Programmed Systems) *Programmed systems:*

- *Cannot perform NCI or CCI operations for concept creation*
- *Cannot reliably answer questions - whether generated by them or by humans - or execute actions except for pure computation*
- *Cannot verify their own outputs except for computational results*
- *Require human intelligence to check reliability in all non-computational tasks requiring understanding any "what", "why" and "how"*

Therefore, all programmed systems fundamentally require human oversight for reliable operation in any task beyond pure computation.

Proof 13 1. *Pure computation refers to:*
   - *Mathematical operations within formal systems*
   - *Logical processing with defined rules*
   - *Data transformations through programmed algorithms*

2. *By logical necessity, programmed systems:*
   - *Cannot perform NCI or CCI operations for concept creation as proved in " 12.3. NCI-CCI Limitation Proof"*
   - *Due to their " 10. Informal Logic Limitation Proof" and " 7. Understanding the Meaning Limitation Proof",*
     - *Cannot reliably answer questions - whether generated by them or by humans - or reliably execute actions except for pure computation*
     - *Cannot reliably verify their own outputs except for computational results*
     - *Cannot reliably perform non-computational tasks requiring understanding any "what", "why" and "how" as it is impossible for them to understand any "what", "why" and "how" in non-computational tasks and, hence, they require human intelligence to check reliability in all non-computational tasks requiring understanding any "what", "why" and "how"*

3. *This limitation applies universally to:*
   - *All current programmed systems*
   - *All future programmed systems*
   - *Any type of programmed system*

12.5 Free Will Limitation Theorem

Theorem 19 (Free Will Limitation Theorem) *Programmed systems cannot possess genuine free will.*

Proof 14 *We proceed by establishing the necessary conditions for free will and showing that programmed systems cannot satisfy these conditions.*

1. ***Definition:*** *Genuine free will is the capacity to make decisions that are not derivable from the existing programming or understanding*
2. ***Premise 1:*** *Free will requires the ability to create concepts not derived from the existing knowledge using New Concept Intuition (NCI) and create concepts derived from the existing knowledge using Combinatorial Concept Intuition (CCI)*
3. ***Premise 2:*** *Free will requires understanding the meaning of one's internal and external environment beyond mere representation*
   - *To make genuine choices, a system must comprehend the significance of options and their consequences and not merely recognize patterns*
   - *Example: Early humans discovering fire from lightning strikes created new concepts about its potential uses (light, warmth, cooking, protection) by understanding the meaning of fire in relation to human needs*
   - *Programmed systems can only represent fire as patterns of data (heat signatures, visual patterns, chemical reactions) without comprehending its meaning*
4. ***Premise 3:*** *Programmed systems cannot have NCI and CCI Capabilities as proven in " 12.3. NCI-CCI Limitation Proof".*
5. ***Premise 4:*** *Programmed systems cannot understand meaning beyond representation as proven in " 7. Understanding the Meaning Limitation Proof".*
   - *Programmed systems operate only at the level of representation, pattern matching and pattern prediction*
   - *They cannot access true meaning, which requires understanding significance beyond representations and their patterns*
6. ***Conclusion:*** *Since programmed systems cannot satisfy the necessary conditions for free will as proved above, they cannot possess genuine free will and any apparent decision-making in programmed systems must ultimately be derivable from their existing programming, regardless of the complexity of that programming or the sophistication of their algorithms*

12.6 Recursive Nature of Knowledge Growth in General Intelligence Systems

When general intelligence systems create new concepts through NCI or CCI operations:

- They become part of the existing knowledge
- Can then be used in CCI operations, enabling further knowledge creation
- Create potential for new NCI insights

This explains how general intelligence can:

- Continuously expand its knowledge
- Create genuinely new understanding through NCI
- Build upon previous insights through CCI
- Achieve continuous growth in capability through NCI-CCI interaction

12.7 Epistemological Implications

12.7.1 NCI-CCI as the Definition of Sentience

NCI-CCI provides the formal epistemological definition of sentience:

Definition 20 (Sentience) *The capability to create and understand new concepts through New Concept Intuition (NCI) and Combinatorial Concept Intuition (CCI) operations.*

This definition:

- Captures the fundamental distinction between mechanical computation and conscious understanding of the meanings of various concepts and capacity for NCI-CCI operations
- Shows why human cognition requires going beyond the existing frameworks
- Explains why programmed systems cannot achieve general intelligence
- Formalizes the unique creative capability of sentient beings
- Provides a verifiable criterion for sentience in any knowledge system based on the NCI-CCI capabilities of human beings as experimentally verified by us

12.7.2 Historical Context in Epistemology

Previous epistemological theories failed to resolve fundamental questions about knowledge creation:

1. Rationalist Tradition:
   - Descartes (1637): Emphasized deductive reasoning but could not explain new concept creation
   - Leibniz (1714): Focused on logical truth but failed to account for genuine insights
   - Limited to manipulating the existing concepts (what we now identify as CCI)
2. Empiricist School:
   - Locke (1689): Reduced knowledge to sensory data, could not explain concepts
   - Hume (1748): Skepticism about causation highlighted the limits of pure empiricism
   - Failed to bridge the gap between experience and novel understanding
3. Kant's (Kant, 1781) Attempted Synthesis:
   - Combined rationalist and empiricist insights
   - Introduced the concept of "synthetic a priori judgments" to explain new concept creation in mathematics and metaphysics
   - Advanced understanding of how the mind structures experience
   - Did not provide a complete formal framework for understanding knowledge creation
4. Modern Epistemological Theories:
   - Offer diverse approaches to knowledge creation and justification
   - Have made progress in specific domains of knowledge analysis
   - Generally focus more on knowledge justification than on the creation of novel concepts
   - Have not yet developed a comprehensive formal framework that fully accounts for the distinction between mechanical and human cognition
   - Did not formally define sentience
   - Continue to face challenges in explaining the emergence of genuinely new concepts

12.7.3 NCI-CCI Framework as a Breakthrough in Epistemology

The NCI-CCI framework provides the first complete resolution of various fundamental questions of epistemology by:

1. Establishing Logical Certainty:
   - Provides logical completeness via excluded middle
   - Creates a rigorous foundation for epistemology
2. Formally Defining Sentience:
   - First precise definition through NCI and CCI capabilities
   - Distinguishes human cognition from mechanical cognition
   - Explains why AGI is impossible
3. The Only Possible Universal Knowledge Creation Theory:
   - Exhaustive categorization through NCI-CCI
   - Unifies insight and derivation
   - No possible third category
4. Resolving Historical Debates:
   - Bridges rationalist-empiricist divide
   - Explains both logical derivation and novel insight
   - Provides formal framework previous theories lacked

This historical context demonstrates the breakthrough nature of the NCI-CCI framework in finally resolving questions that have challenged epistemology for centuries.

12.8 Additional Implications of NCI-CCI Framework

12.8.1 Nature of Human Awareness

The NCI-CCI framework reveals fundamental aspects of human cognition:

- NCI and CCI capabilities imply continuous potential for creating new concepts
- Explains the unbounded creative capability of human cognition
- Shows why human cognition cannot be reduced to computational processes

12.8.2 Scientific Progress

It explains the mechanism of scientific advancement:

- Major breakthroughs often require NCI (new fundamental concepts, as seen in general relativity by Einstein (1916) conceptualizing gravity as the curvature of spacetime, which transcended all the existing concepts of gravity and could not be derived from them)
- Minor breakthroughs often operate just through CCI (combining the existing concepts)
- Explains why machines can assist but cannot replace human scientists

12.8.3 Mathematical Understanding

It illuminates the nature of mathematical insight:

- New mathematical concepts get created through NCI as well as CCI
- Proofs and derivations operate through the combination of CCI and NCI (NCI also as seen in breakthroughs like the proof of Poincaré conjecture by Perelman (2002) requiring new Ricci flow insights and the proof of Fermat's Last Theorem by Wiles (1995) requiring novel connections between modular forms and elliptic curves)
- Explains why automated theorem provers cannot replace human mathematicians

12.8.4 Artistic Creation

It provides insight into artistic creativity:

- Genuine artistic innovation requires NCI
- Derivative works operate through CCI
- Explains why AI art tools can combine but cannot truly create

12.8.5 Educational Theory

It has implications for learning and education:

- True innovation requires developing NCI and CCI capabilities
- Mere memorization is not understanding the meaning of a concept or its instance
- Suggests focus on fostering understanding and NCI and CCI capabilities over mechanical learning

12.8.6 Language and Meaning

It reveals the nature of linguistic understanding:

- Creation of new meanings requires NCI
- Combination of the existing meanings is CCI
- Explains why language models manipulate but do not understand the meaning of words/phrases/sentences/paragraphs/logical structure

12.8.7 Cultural Evolution

It explains the mechanisms of cultural development:

- Cultural innovations require NCI and CCI
- Cultural transmission operates through CCI
- Shows why cultural evolution needs human creativity

13 Results and Discussion

13.1 The Logical Impossibility of the Turing Test

This section demonstrates how our proof shows the fundamental impossibility of the Turing test itself.

13.1.1 Analysis of the Turing Test

The Turing test, proposed by Turing (1950), suggests that a machine could be considered exhibiting intelligent behaviour equivalent to that of a human if it could imitate human conversational behavior convincingly.

13.1.2 Fundamental Problems

The test is logically impossible because:

1. Due to various fundamental limitations proven in our 8 AGI Logical Impossibility proofs, with 7 independent proofs and the 8th proof being based on 4 independent proofs but with its independent importance
2. Each of these 7 limitations independently makes AGI logically impossible
3. Therefore, each of these 7 independent proofs independently makes the Turing test meaningless, making it impossible for any machine to truly pass a test for general intelligence

13.1.3 The Simulation Fallacy

The test falls prey to a fundamental fallacy:

- Assumes simulating behavior equals capability
- Ignores the necessity of various aspects of general intelligence as explained in 7 independent proofs
- Confuses pattern manipulation with understanding
- Mistakes programmed formal system operations for general intelligence

13.1.4 Impact

This understanding transforms:

- Countless research programs aimed at passing the Turing test
- AI field's fundamental assumptions
- Basic measures of AI progress

13.1.5 Implications

Our proof shows the Turing test is:

- Testing for a logically impossible property
- Like testing if triangles can be circles
- Fundamentally meaningless
- Based on a deep misconception about general intelligence being similar to formal systems

13.1.6 Proper Testing Framework

This implies AI should be tested:

- Only for specific computational capabilities

- Within fundamental programmed system limitations without claims of “thinking” or “understanding” or “exhibiting intelligent behaviour equivalent to that of a human"
- Acknowledging these boundaries as fundamental rather than temporary

The Turing test, thus, joins perpetual motion and square circles in the category of logically impossible propositions. This demonstrates how our proofs not only resolve the question of AGI's logical impossibility but also show the fundamental flaws in how we have been conceptualizing and testing machine intelligence.

13.2 Implications

13.2.1 For Computer Science and Artificial Intelligence

This proof establishes that:

- AI faces multiple fundamental limitations as established in our 8 proofs
- These limitations are:
  - Logically proven to be insurmountable
  - Independent of technological progress
  - Fundamental to the nature of programmed systems
- Therefore:
  - AGI is logically impossible
  - AI development should focus on specific, limited domains
  - Success criteria should be defined within these limitations
  - Focus should be on adding new "NCI and CCI"-based knowledge of human beings dynamically in the AI training algorithm
  - Clear boundaries exist for AI development
  - Human-AI collaboration is the optimal path

13.2.2 For Cognitive Science

- Knowledge creation in all knowledge systems has a universal NCI-CCI framework
- Gaps between the levels of general intelligence and AI cannot be bridged by any type and level of computation by an AI or programmed system
- Sentience is beyond any programmed system due to NCI-CCI limitation

14 Conclusions

Thus, we give 8 proofs - with 7 independent proofs and the 8th proof being based on 4 independent proofs but with its independent importance - of the logical impossibility of Artificial General Intelligence (AGI) while providing a comprehensive framework for understanding the nature of general intelligence and knowledge creation, and the limitations of programmed systems. We also provide detailed pointers for future AI research.

References

René Descartes. *Discourse on the Method of Rightly Conducting One's Reason and of Seeking Truth in the Sciences*. Jan Maire, Leiden, Netherlands, 1637.

Albert Einstein. The foundation of the general theory of relativity. *Annalen der Physik*, 354:769–822, 1916. doi: 10.1002/andp.19163540702. English translation available in “The Principle of Relativity" book (1923).

Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931. doi: 10.1007/BF01700692.

David Hume. *An Enquiry Concerning Human Understanding*. Andrew Millar, London, 1748.

Immanuel Kant. *Critique of Pure Reason*. Johann Friedrich Hartknoch, Riga, 1781.

S. C. Kleene. On notation for ordinal numbers. *Journal of Symbolic Logic*, 3(4):150—155, 1938. doi: 10.2307/2267778.

Gottfried Wilhelm Leibniz. *The Monadology*. 1714.

John Locke. *An Essay Concerning Human Understanding*. Thomas Basset, London, 1689.

Grisha Perelman. The entropy formula for the Ricci flow and its geometric applications. *arXiv preprint math/0211159*, 2002. URL `https://arxiv.org/abs/math/0211159`.

Alfred Tarski. The concept of truth in formalized languages. In John Corcoran (ed.), *Logic, Semantics, Metamathematics: Papers from 1923 to 1938*. Hackett Publishing Company, 1983. Originally published as "Der Wahrheitsbegriff in den formalisierten Sprachen", Studia Philosophica, 1936.

A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950. doi: 10.1093/mind/LIX.236.433.

Andrew Wiles. Modular elliptic curves and Fermat's last theorem. *Annals of Mathematics*, 141(3):443–551, 1995. doi: 10.2307/2118559.