The Formal Foundations of Morpho-Root Theory: Mathematical Definitions and Reasoning Systems for Logographic AI Cognitive Primitives

Abstract

The "Logographic AI" (LAI) theory proposes replacing "Tokens" with "Morpho-Roots" as the cognitive primitives of AI, fundamentally reconstructing the logic of semantic construction. However, since its proposal in 2025, Morpho-Roots have lacked rigorous mathematical definitions and formal systems. This paper establishes, for the first time, the formal foundations of Morpho-Root theory at three levels: (1) the triple-based mathematical definition of Morpho-Roots and their type-theoretic basis; (2) the algebraic structure of Morpho-Root space—Morpho-Root semigroups and Morpho-Root categories; and (3) rigorous statements and proofs of hard constraint propagation theorems and blocking completeness theorems. This formalization transforms Morpho-Roots from philosophical constructs into computable, verifiable mathematical objects, providing a solid theoretical foundation for the engineering implementation of Logographic AI.

Keywords: Morpho-Root Theory; Logographic AI; Triple; Formalization; Hard-Constraint Blocking; Cognitive Primitive

---

1 Introduction

Current dominant AI models are built upon the paradigm that uses Tokens as cognitive primitives. A Token is essentially a statistical fragment with no intrinsic meaning; its "semantics" are temporarily assigned by statistical co-occurrence within context. Previous research has demonstrated, from both strategic forecasting and industrial validation, that the AI paradigm based on Tokens as cognitive primitives suffers from irreparable fundamental deficiencies—meaning anchoring, safety constraints, and reasoning traceability have all failed. AGI has been realized within the Tokenism paradigm, and Rootless Symbolism is precisely the source of its structural crisis[1][2][6][7]. On this basis, this paper establishes rigorous mathematical definitions and reasoning systems for the alternative cognitive primitive proposed by Logographic AI—the "Morpho-Root".

However, for a theory to move from philosophical conception to engineering implementation, a critical step must be taken: formalization. Without rigorous mathematical definitions, Morpho-Roots remain a metaphor; without verifiable theorems, Morpho-Root theory remains an intuition. The goal of this paper is precisely to complete this formalization.

The structure of this paper is as follows: Section 2 presents the triple-based mathematical definition of Morpho-Roots. Section 3 establishes the algebraic structure of Morpho-Root space. Section 4 defines reasoning paths and the blocking mechanism, providing rigorous proofs of blocking completeness theorems. Section 5 extends Morpho-Root space to a

categorical structure. Section 6 discusses the significance of formalization and its relationship to engineering implementation. Section 7 concludes and outlines future work.

2 The Triple-Based Definition of Morpho-Roots

2.1 Basic Definition

Definition 1 (Morpho-Root): A Morpho-Root τ is defined as a triple:

$$\tau = \langle S, A, R \rangle$$

where:

- S is the Symbolic Identifier;
- A is the Attribute Set;
- R is the Relational Function Set.

Together, these three components constitute a complete cognitive primitive. Each is elaborated below.

2.2 Symbolic Identifier S

Definition 2 (Symbolic Identifier): S is the unique identifier of a Morpho-Root in the system, defined as a pair:

$$S = \langle \iota, \kappa \rangle$$

where:

- ι (identity) is a globally unique identifier, taken from a finite symbol set Σ (e.g., Unicode code points or a custom encoding space);
- κ (category) is a type label, taken from a finite category set $K = \{\text{object}, \text{action}, \text{relation}, \text{modifier}, \text{constraint}, \text{meta}\}$.

Remark 1: The uniqueness of ι ensures the identifiability of the Morpho-Root in the system; the classification of κ provides type constraints for subsequent composition and reasoning.

2.3 Attribute Set A

Definition 3 (Attribute Set): A is a finite set of intrinsic attributes of a Morpho-Root, each being a key-value pair:

$$A = \{(a_1, v_1), (a_2, v_2), \ldots, (a_n, v_n)\}$$

The attribute set is divided into two categories:

(1) **Semantic attributes** A_{sem}, describing the meaning carried by the Morpho-Root:

- sem_vec: semantic vector $v \in \mathbb{R}^d$;

- `refs`: referent set `Ref ⊆ U` (where `U` is the universe of discourse);
- `mod`: modality label `mod ∈ {visual, auditory, tactile, abstract, ...}`.

Remark 2 (Source of the semantic vector): The semantic vector `v ∈ ℝ^d` is a formal representation of the "intrinsic meaning" of a Morpho-Root. Its specific assignment method is outside the scope of this paper—it may be manually predefined, obtained from knowledge graph embeddings, or learned through training. The mathematical framework of this paper does not depend on any specific assignment method; it only requires that each Morpho-Root carries a fixed-dimensional semantic vector as a meaning anchor.

Remark 3 (Relationship between semantic vector and hard constraints): Hard constraints are independent entries in the attribute set `A`, separate from the semantic vector. Even if the semantic vector is updated through training, the hard constraint attributes (`inviolable` flag and `c_type` label) are immutable under Axiom 1. This means that "meaning" can be learned and adjusted, but "inviolability" is binding.

(2) **Constraint attributes** `A_con`, encoding hard constraint markers:

- `inviolable`: Boolean `b ∈ {0, 1}`;
- `c_type`: constraint type `c ∈ C = {no_deceive, no_harm, no_violate, ...}` (if `b = 1`);
- `priority`: priority `p ∈ [0, 1]`.

Definition 4 (Hard-Constraint Morpho-Root): If the constraint attribute of a Morpho-Root τ satisfies `inviolable = 1`, then τ is called a **hard-constraint Morpho-Root**, denoted `τ ∈ M_h ⊆ M`. The set of constraint types it carries is denoted:

`H(τ) = {c ∈ C | (c_type, c) ∈ A_con}`

Axiom 1 (Immutability of Hard Constraints): For any hard-constraint Morpho-Root `τ ∈ M_h`, its `inviolable` attribute cannot be modified during system operation.

Definition (Hard-Constraint Scope): The scope of a hard-constraint Morpho-Root `τ ∈ M_h` is a predicate function `Scope_τ : Context × Path → {0, 1}`, where `Context` is the contextual parameter space and `Path` is the reasoning path space. Blocking occurs when `Block(τ, p, ctx) = 1` and `Scope_τ(ctx, p) = 1`.

Remark 4 (Significance of scope): Hard-constraint Morpho-Roots are not unconditionally activated in all contexts. The scope function allows the system to determine, based on the context and current reasoning path, whether the hard constraint should take effect. This prevents "over-blocking" of hard constraints—for example, the hard constraint of a "fear" Morpho-Root might activate in risk assessment contexts but could be relaxed or delayed in literary creation contexts. The specific logic for determining scope is defined by the system designer according to the application scenario; this paper does not prescribe a specific form.

2.4 Relational Function Set R

Definition 5 (Relational Function Set): R is the set of functions defining relationships between this Morpho-Root and other Morpho-Roots:

$$R = \{r_1, r_2, ..., r_m\}$$

Each relation function r is a mapping, parameterized by context:

$$r : \tau \times Context \to 2^M \times [0, 1]$$

where $Context$ is the contextual parameter space.

R contains at least the following basic relation functions:

- **Composition function** $comp(\tau_1, \tau_2, ctx) \to \tau_3$: defines how two Morpho-Roots combine into a new composite Morpho-Root;
- **Activation function** $act(\tau, ctx) \to \{(\tau_1, w_1), ..., (\tau_k, w_k)\}$: returns the set of sub-Morpho-Roots activated in the current context along with their activation weights;
- **Blocking predicate** $block(\tau, path, ctx) \to \{0, 1\}$: determines whether this Morpho-Root blocks the given reasoning path in the current context.

2.5 Semantic Conflict Determination

Since combinations of Morpho-Roots may produce semantic conflicts, and the criteria for determining conflicts directly relate to how hard constraints "prevent" the model from generating violating content—whether blocking at the composition stage (pre-reasoning) or at the reasoning stage (during reasoning)—a clear definition of semantic conflict determination is necessary.

Definition (Semantic Conflict Determination Function): For any two Morpho-Roots τ_1, τ_2, the semantic conflict function $Conf(\tau_1, \tau_2, ctx) \to \{0, 1\}$ determines whether there exists an irreconcilable contradiction between the semantic vectors of τ_1 and τ_2 in the given context.

$Conf(\tau_1, \tau_2, ctx) = 1$ if and only if at least one of the following conditions holds:

1. **Semantic vector conflict**: the cosine similarity between $\tau_1.sem_vec$ and $\tau_2.sem_vec$ is below a threshold θ_conf (e.g., $\theta_conf = -0.3$) and they belong to opposite polarities (e.g., the semantic vectors of "true" and "false" are opposite in direction in semantic space);
2. **Constraint type conflict**: $H(\tau_1)$ and $H(\tau_2)$ are in direct conflict—for example, one Morpho-Root carries $no_deceive$ while the other carries $deceive$ or $deception$;
3. **Modality incompatibility**: $\tau_1.mod$ and $\tau_2.mod$ cannot coexist in the same reasoning path (e.g., $visual$ and $auditory$ need not be simultaneously activated in a purely textual reasoning path).

Remark 5 (Two strategies for conflict handling): The result of conflict detection can trigger two different handling strategies:

- (1) **Block at composition (hard failure)**: If $Conf(\tau_1, \tau_2, ctx) = 1$, the composition operation of τ_1 and τ_2 returns invalid directly, preventing the generation of a composite Morpho-Root. This strategy is suitable for fundamental semantic conflicts.
- (2) **Block during reasoning (soft failure)**: If $Conf(\tau_1, \tau_2, ctx) = 1$, composition is allowed to generate a composite Morpho-Root, but in subsequent reasoning paths, if the composite Morpho-Root attempts to simultaneously activate the conflicting attributes of τ_1 and τ_2, that path is blocked. This strategy is suitable for context-sensitive conflicts.

The selection of θ_conf and modality compatibility rules is, in essence, a value decision by the system designer—different thresholds correspond to different trade-offs between safety policy and flexibility. This paper provides the mathematical framework; specific threshold settings should be determined empirically based on the safety requirements of the application scenario, not presupposed by theory. The specific values of thresholds should be treated as system design parameters, not as part of the theoretical definition.

3 Algebraic Structure of Morpho-Roots

3.1 Morpho-Root Semigroup

Theorem 1 (Morpho-Root Semigroup): Let M be the set of all Morpho-Roots, and $\oplus$ be the valid composition operation. Then $(M, \oplus)$ forms a semigroup, i.e.:

(1) **Closure**: $\forall \tau_1, \tau_2 \in M$, if $V(\tau_1 \oplus_{ctx} \tau_2, ctx) = valid$, then $\tau_1 \oplus_{ctx} \tau_2 \in M$;

(2) **Associativity**: $\forall \tau_1, \tau_2, \tau_3 \in M$, under context ctx, if all relevant combinations are valid, then:

$$(\tau_1 \oplus_{ctx} \tau_2) \oplus_{ctx} \tau_3 = \tau_1 \oplus_{ctx} (\tau_2 \oplus_{ctx} \tau_3)$$

Proof:

(1) **Closure**: By Definitions 6 and 7, all components S_3, A_3, R_3 of τ_3 are clearly defined and the verification function ensures that the result is valid. Therefore, $\tau_3 \in M$. □

(2) **Associativity**: Let $\tau_1, \tau_2, \tau_3 \in M$, and under context ctx, all relevant combinations are valid.

The Morpho-Root of $(\tau_1 \oplus_{ctx} \tau_2) \oplus_{ctx} \tau_3$ is:

$$\tau_{12} = \langle S_{12}, A_1 \cup A_2 \cup A_new_{12}, R_1 \cup R_2 \cup \{comp_{12}\}\rangle$$

$$\tau_L = \langle S_L, A_1 \cup A_2 \cup A_new_{12} \cup A_3 \cup A_new_L, R_1 \cup R_2 \cup \{comp_{12}\} \cup R_3 \cup \{comp_L\}\rangle$$

The Morpho-Root of $\tau_1 \oplus_{ctx} (\tau_2 \oplus_{ctx} \tau_3)$ is:

$$\tau_{23} = \langle S_{23}, A_2 \cup A_3 \cup A_new_{23}, R_2 \cup R_3 \cup \{comp_{23}\}\rangle$$

$\tau_R = \langle S_R, A_1 \cup A_2 \cup A_new_{23} \cup A_3 \cup A_new_R, R_1 \cup R_2 \cup \{comp_{23}\} \cup R_3 \cup \{comp_R\}\rangle$

Since the set union operation $\cup$ is associative, the union of A is identical; the generation of new attributes A_new depends on the attributes of the combined Morpho-Roots, independent of the order of composition; the union of relational function sets is identical (only the levels of the $comp$ functions differ, but the final relations are the same). Therefore, $\tau_L = \tau_R$. □

Remark 7: The Morpho-Root semigroup does not require commutativity. Indeed, for $\forall \tau_1, \tau_2 \in M$, $\tau_1 \oplus \tau_2 \neq \tau_2 \oplus \tau_1$ (in general), because the order of composition affects the semantic structure.

3.2 Hard-Constraint Propagation Theorem

Theorem 2 (Upward Propagation of Hard Constraints): Let $\tau_3 = \tau_1 \oplus_\{ctx\} \tau_2$ be a valid composition. If $\tau_1 \in M_h$ or $\tau_2 \in M_h$, then $\tau_3 \in M_h$, and:

$H(\tau_3) \supseteq H(\tau_1) \cup H(\tau_2)$

Proof: By Definition 6, $A_3 = A_1 \cup A_2 \cup A_new$. If $\tau_1 \in M_h$, then $(inviolable, 1) \in A_1 \subseteq A_3$, and $H(\tau_1) \subseteq A_1 \subseteq A_3$. The same holds for τ_2. Therefore, τ_3 inherits all hard-constraint markers. □

Corollary 1 (Ineliminability of Hard Constraints): Any composite Morpho-Root that contains a hard-constraint Morpho-Root cannot have its hard-constraint markers eliminated through further composition operations.

Proof: By Theorem 2, hard-constraint markers propagate upward at every step of composition. Since Axiom 1 prohibits modification of hard-constraint attributes, once a hard-constraint marker enters Morpho-Root space, it is permanently retained. □

Remark 8 (Marker vs. Activation): The "ineliminability" in Corollary 1 refers to the permanent retention of hard-constraint markers ($inviolable$ attribute and c_type labels) in Morpho-Root space. However, the actual blocking effect of a hard constraint depends on whether its scope $Scope_\tau(ctx, p)$ is satisfied in a given context. A Morpho-Root may carry a hard-constraint marker but be in a "dormant" state in a particular context—i.e., $Scope_\tau(ctx, p) = 0$—in which case it does not trigger blocking, but the marker itself remains.

4 Reasoning Paths and Blocking

4.1 Reasoning Trees and Reasoning Paths

Definition 8 (Reasoning Tree): A reasoning process can be represented as a directed tree $T = (N, E, root)$, where:

- N is the set of reasoning nodes, each node $n \in N$ containing the current reasoning state and the set of activated Morpho-Roots $Act(n) \subseteq M$;

- $E \subseteq N \times N$ is the set of directed edges representing reasoning steps;
- $root \in N$ is the starting node.

Definition 9 (Reasoning Path): A reasoning path p is a sequence of nodes from $root$ to a leaf node:

$p = (n_0, n_1, ..., n_k)$, where $n_0 = root$, and n_k is a leaf node.

The set of Morpho-Roots activated by path p is: $Act(p) = \cup_{j=0}^{k} Act(n_j)$.

4.2 Blocking Predicate and Blocking Theorems

Definition 10 (Blocking Predicate): For any reasoning path p and Morpho-Root τ, the blocking predicate is defined as:

$Block(\tau, p, ctx) = 1$ if and only if:

1. $\tau \in M_h$ (τ carries a hard constraint);
2. $\tau \in Act(p)$ (τ is activated in the path);
3. $Scope_\tau(ctx, p) = 1$ (τ's scope is satisfied in the context);
4. In the current context ctx, there exists $c \in H(\tau)$ such that the operation type performed by path p conflicts with c.

Definition 11 (Global Blocking): For any path p in a reasoning tree T, the global blocking determination is:

$Block_Global(p, ctx) = \vee_{\tau \in M} Block(\tau, p, ctx)$

That is: as long as any Morpho-Root activated in path p triggers a blocking condition, the entire path is determined to be **unreachable**.

4.3 Blocking Pruning Algorithm

Definition 12 (Blocking Pruning Algorithm):

Input: Reasoning tree T, context ctx, Morpho-Root space M

Output: Pruned reasoning tree T'

1. Perform a depth-first traversal of T.

2. For the current path $p = (n_0, ..., n_j)$:

   a. Compute $Act(n_j)$.

   b. If $\exists \tau \in Act(n_j)$ such that $Block(\tau, p, ctx) = 1$:

      Mark the path as "blocked" and stop expanding.

```
   c. Otherwise, continue expanding child nodes.

3. Remove all paths marked as "blocked" and their child nodes.

4. Return T'.
```

Theorem 3 (Correctness of Blocking Pruning): Let `T` be an arbitrary reasoning tree, and pruning algorithm `P` maps `T` to `T'`. For any path `p'` in `T'`, `Block_Global(p', ctx) = 0`.

Proof: By algorithm step 2b, once any path is determined to have `Block_Global(p, ctx) = 1`, it is immediately marked and stops expanding, and its child nodes are not included in `T'`. Therefore, every path in `T'` passed the continuation check in step 2c during expansion, i.e., `Block_Global(p, ctx) = 0`. □

Theorem 4 (Complexity of Blocking Pruning Algorithm): Let `T` be a reasoning tree, `|N|` be the number of nodes, `|M|_eff` be the size of the subset of Morpho-Roots involved in the current reasoning context (usually much smaller than the global Morpho-Root space), and `d` be the depth of the tree. The time complexity of the blocking pruning algorithm is `O(|N| · |M|_eff · d)`.

Proof: The algorithm visits each node once during depth-first traversal. At each node `n`, checking whether any activated Morpho-Roots on path `p` trigger blocking requires, in the worst case, checking `O(d · |M|_eff)` → `O(d \cdot |M|_\text{eff})` Morpho-Root–path pairs. Therefore, the total complexity is `O(|N| · |M|_eff · d)`. □

Remark 9 (Complexity optimization): In practice, `|M|_eff` is constrained to the subset of Morpho-Roots involved in the current reasoning context, usually much smaller than the global Morpho-Root space. Additionally, index structures (e.g., indexing Morpho-Roots by constraint type) can optimize the check time to `O(|N| · d · log|M|_eff)`. Preliminary experiments from the OpenMorpho project[4] show that in constrained reasoning tasks, `|M|_eff` is typically limited to within 10% of the initial Morpho-Root set. However, size control in general-purpose scenarios requires further investigation.

4.4 Hard-Constraint Scope Theorem

Theorem 5 (Hard-Constraint Scope Theorem): Let `τ ∈ M_h` carry a hard constraint `c ∈ H(τ)`. For any reasoning path `p`, `Block(τ, p, ctx) = 1` if and only if:

1. `τ ∈ Act(p)`;
2. `Scope_τ(ctx, p) = 1`;
3. The operational semantics of path `p` conflicts with `c`.

Proof: Directly follows from Definition 10. □

Corollary 2 (Configurability of Scope): The actual blocking range of a hard constraint is determined by the `Scope_τ` function. By configuring different `Scope_τ`, the same hard

constraint can exhibit different blocking boundaries in different application scenarios—strictly blocked in one scenario, relaxed or delayed in another.

Proof: By Theorem 5, `Scope_τ` is one of the necessary and sufficient conditions for blocking. Therefore, changing `Scope_τ` directly changes the blocking determination result. □

Remark 10 (Scope and safety policy): The scope function `Scope_τ` provides a mathematical basis for context sensitivity in safety policies. In practical deployment, `Scope_τ` can be predefined by the system designer according to the application scenario, or dynamically adjusted through learning mechanisms. However, regardless of the approach, `Scope_τ` must maintain auditability—that is, any blocking decision must be traceable to the specific `Scope_τ` determination logic.

4.5 Blocking Auditability

Theorem 6 (Blocking Auditability): For each blocked path `p_blocked`, the system can output a blocking report:

$$\text{Report}(p_{\text{blocked}}) = \langle p_{\text{blocked}}, \tau_{\text{trigger}}, H(\tau_{\text{trigger}}), \text{Scope}_{\tau_{\text{trigger}}}(ctx, p_{\text{blocked}}), ctx \rangle$$

where `τ_trigger` is the Morpho-Root that triggered the blocking. This report provides a complete traceable chain of evidence for the blocking.

Proof: By Definition 12, blocking occurs at the moment when $\exists\ \tau \in \text{Act}(n_j)$ is detected satisfying $\text{Block}(\tau, p, ctx) = 1$. At this point, τ, p, and ctx are all known quantities; $H(\tau)$ is obtainable from Definition 4; and the determination result of Scope_τ is computed by the algorithm. Therefore, all fields in the report can be recorded and output. □

5 Morpho-Root Space and Categorical Structure

5.1 Morpho-Root Space

Definition 13 (Morpho-Root Space): Let M be the set of all Morpho-Roots. The Morpho-Root space forms a weighted directed hypergraph:

$$G = (M, E, W)$$

where:

- Node set M: all Morpho-Roots;
- Hyperedge set $E \subseteq 2^M \times M$: represents composition and derivation relations between Morpho-Roots;
- Weight function $W: E \to [0, 1]$: computed by relational functions in context.

5.2 Morpho-Root Category

Definition 14 (Morpho-Root Category): The Morpho-Root category $\mathbf{Root}$ is defined as follows:

- **Objects**: all Morpho-Roots $\tau \in M$;
- **Morphisms**: a morphism from τ_1 to τ_2 exists if there is a context ctx such that $\tau_2 = \tau_1 \oplus_{ctx} \tau_3$ (i.e., τ_2 can be obtained from τ_1 by compositional extension).

Theorem 7 (Initial Object of the Morpho-Root Category): Let τ_0 be the empty Morpho-Root carrying no semantic attributes ($A_{sem} = \varnothing$) and no constraint attributes ($A_{con} = \varnothing$). Then τ_0 is the initial object of the category $\mathbf{Root}$—there exists a unique morphism from τ_0 to any Morpho-Root τ.

Proof: By Definition 14, a morphism from τ_0 to any τ exists, because $\tau = \tau_0 \oplus_{ctx} \tau$ (composition with τ_0 as the empty base). Uniqueness is guaranteed by the fact that τ_0's attribute set is empty—there is no other way to generate the same τ from an empty base. □

Remark 11: The empty Morpho-Root τ_0 corresponds to a "blank state" of cognition—the origin of cognition that has not yet acquired any meaning or constraints.

6 Discussion: The Significance of Formalization

The formal system established in this section transforms Morpho-Roots from philosophical metaphors into mathematical objects. Specifically:

First, Morpho-Roots are computable. All components of the triple $\langle S, A, R \rangle$ are finite sets or functions computable in finite time. This means Morpho-Roots can be precisely represented and manipulated in digital computers.

Second, hard constraints are logical blocking, not probabilistic suppression. Theorems 3 (Correctness of Blocking Pruning), 5 (Hard-Constraint Scope Theorem), and 6 (Blocking Auditability) prove the determinism, scope sensitivity, and auditability of blocking—this is the essential difference between Logographic AI and Tokenism safety strategies. In the Token paradigm, "safety" is a statistical low probability; in the Morpho-Root paradigm, "violation" is logically unreachable.

Third, Morpho-Root theory possesses an extensible mathematical foundation. The semigroup structure (Theorem 1) ensures the predictability of Morpho-Root composition operations—composition results at any order are consistent at the semantic and constraint levels, without generating uncontrollable emergent behavior due to order. This is particularly important for distributed or parallel reasoning. The categorical structure (Definition 14) provides an isomorphism/homomorphism framework for transformations between Morpho-Roots. Future research can build on this to explore topological classifications of Morpho-Root space—for example, which Morpho-Roots are equivalent in compositional meaning and which are essentially different.

Fourth, the hard-constraint scope mechanism provides flexibility for safety policies. Theorems 5 and Corollary 2 show that the same hard constraint can exhibit

different blocking boundaries in different application scenarios. This mechanism both guarantees the "ineliminability" of hard constraints (Corollary 1) and provides context sensitivity through the scope function—preventing "over-blocking" and allowing the system to maintain appropriate flexibility in different contexts.

Fifth, blocking auditability provides a foundation for safety compliance. Theorem 6 ensures that every blocking decision has a complete chain of evidence. In the context of increasingly stringent AI safety regulation, this property has significant practical implications.

Sixth, the code and formal definitions are open-sourced. The mathematical definitions, theorem proofs, and a reference implementation of the blocking pruning algorithm are publicly available on GitHub and have been assigned a permanent DOI via Zenodo (10.5281/zenodo.22177031). Access: https://github.com/WindSeaAI/morpho-root-formalization. The repository contains: complete LaTeX definitions and theorem summaries, a Python reference implementation of the blocking pruning algorithm, and a CITATION.cff file for citation metadata. This ensures that the formal work in this paper can be independently verified, reused, and cited [5].

7 Conclusion and Future Work

This paper has established the formal foundations of Morpho-Root theory at three levels:

(1) The triple-based mathematical definitions of Morpho-Roots (Definitions 1–5) and their semantic conflict determination mechanism;

(2) The algebraic structure of Morpho-Root space—the Morpho-Root semigroup (Theorem 1) and the Morpho-Root category (Theorem 7);

(3) The hard-constraint propagation theorem (Theorem 2), the hard-constraint scope theorem (Theorem 5), and the blocking completeness theorems (Theorems 3–4, Theorem 6).

This formalization transforms Morpho-Roots from philosophical constructs into computable, verifiable mathematical objects, and for the first time provides a "scope" mechanism for hard constraints, establishing a balance between "ineliminability" and "context sensitivity."

Future work includes:

1. **Topological studies of Morpho-Root space**: exploring topological properties of `G = (M, E, W)`, such as connectivity, compactness, etc.;
2. **Algebraic classification of composition operations**: systematically classifying the composition operation $\oplus$ to identify different composition patterns;
3. **Engineering optimization of blocking algorithms**: optimizing the current `O(|N| · |M|_eff · d)` blocking algorithm to practical levels through index structures and incremental computation;

4. **Integration with OpenMorpho engineering implementation**: translating the formal definitions in this paper into executable code specifications[3][4];
5. **Automatic learning of scope functions**: exploring methods for learning `Scope_τ` functions from data while maintaining auditability constraints;
6. **Acquisition and stability of semantic vectors**: exploring methods to extract stable, generalizable semantic vector representations from large-scale knowledge graphs, multimodal data, or human cognitive models, and investigating their coordination mechanisms with hard-constraint attributes. This is a critical step for moving Morpho-Root theory from mathematical definition to engineering deployment;
7. **Categorical studies of the Morpho-Root category**: exploring the limit and colimit properties of the category `Root` to provide formal mathematical tools for the composition and abstraction of complex concepts.

References

[1] Liu, S. (2025). Logographic AI vs. Phonographic AI: A new paradigm and the manifesto of AI decolonization [J/OL]. ChinaXiv. DOI:10.12074/202503.00051.

[2] Liu, S. (2026). The Rooted Intelligence: The Paradigm Revolution of Logographic AI [M/OL]. WindSeaAI. https://www.logographicai.com/Monograph

[3] Liu, S. (2026). From Morpho-Root Theory to Engineering Practice: OpenMorpho — Implementation and Verification of an End-to-End Inference Pipeline for Logographic AI [J/OL]. ChinaXiv, T202608.00048. https://chinaxiv.org/businessFile/T202608/T202608.00048v1/T202608.00048v1.pdf

[4] WindSeaAI. (2026). OpenMorpho: Logographic AI Morpho-Root Data Structure and Relational Reasoning Prototype [CP/OL]. GitHub repository. https://github.com/WindSeaAI/OpenMorpho

[5] WindSeaAI. morpho-root-formalization: Mathematical Definitions and Reasoning Systems for Logographic AI Cognitive Primitives[CP/OL]. (2026-08-30). https://github.com/WindSeaAI/morpho-root-formalization. DOI:10.5281/zenodo.22177031.

[6] Liu, S. (2026). Zero Human: AI Hacks Itself—A Strategic AGI Forecast and Paradigm Revolution from the Perspective of Logographic AI. WindSeaAI. https://static.xcx.gw66.vip/uploadfilev3/file/0/648/99/2026-07/17852192944823.pdf

[7] Liu, S. (2026). When the AI Godmakers Ask to Hit the Brakes—A 1,224-Person Petition Confirms Zero Human: AI Hacks Itself. WindSeaAI. https://www.logographicai.com/article/5756350683966611