The UNNS Many-Faces Theorem: Formalization and Proof Sketches

Research Note

Abstract

We propose and sketch a formal theorem—the "Many-Faces Theorem"—for the UNNS framework (Unbounded Nested Number Sequences / Universal Network Nexus System). The result shows how linear recurrence sequences, attractors, modular domains, and cross-domain homomorphisms are all naturally embeddable in UNNS. The theorem justifies the phrase "many faces" as algebraic, geometric, topological, number-theoretic, and computational perspectives coexist under a single substrate.

1 Definitions

Definition 1 (Nest Set)**.** *Let S be a set of* nests*, i.e., symbolic tokens or integer values.*

Definition 2 (Combinator Algebra)**.** *Let $\mathcal{C}$ be a finite set of operators (combinators) $\star : S^k \to S$. Each $\star$ is specified by a finite algorithm using integer arithmetic, addition, multiplication, shifting (index arithmetic), and modular arithmetic.*

Definition 3 (Generators and Dynamics)**.** *A UNNS configuration is $U = (S, \mathcal{C}, \mathcal{G})$ with $\mathcal{G} \subset S$ as a finite seed set. An update rule applies combinators iteratively to produce new nests.*

Definition 4 (Domain Mappings)**.** *A* domain *D is any mathematical structure (ring, metric space, manifold, graph). A mapping $\mu_D : S \to D$ is* computable *if an algorithm produces $\mu_D(s)$ from a finite description of s. Typical domains:*

- *A: algebraic (polynomials, vectors)*
- *G: geometric ($\mathbb{R}^2$ polar coordinates)*
- *T: topological (graphs)*
- *$\mathbb{Z}_m$: modular residues*

Definition 5 (UNNS System)**.** *A UNNS system is a tuple $(S, \mathcal{C}, \mathcal{G}, \{\mu_D\}_{D \in \mathcal{D}})$ with a finite set of domains $\mathcal{D}$ and computable domain mappings.*

Assumptions. We assume: (1) each combinator is definable by a finite algorithm with bounded lookback; (2) each μ_D is computable and respects linearity of combinators.

2 The Many-Faces Theorem

Theorem 1 (UNNS Many-Faces Theorem). *Let $U = (S, \mathcal{C}, \mathcal{G}, \{\mu_D\}_{D\in\mathcal{D}})$ be a UNNS system under the above assumptions. Then:*

1. ***Linear recurrence embedding.*** *Any linear recurrence $a_n = c_1 a_{n-1} + \cdots + c_r a_{n-r}$ can be embedded: there exists $s_n \in S$ such that $\mu_{\mathbb{Z}}(s_n) = a_n$.*

2. ***Dominant-root attractor.*** *If the characteristic polynomial has a unique dominant root r_d, then $\lim_{n\to\infty} \mu_{\mathbb{R}}(s_{n+1})/\mu_{\mathbb{R}}(s_n) = r_d$. Geometric embedding μ_G yields logarithmic spirals with growth parameter $\ln |r_d|$.*

3. ***Modular domain partition.*** *For any modulus m, residues $\mu_{\mathbb{Z}_m}(s_n)$ partition nests into m domains, evolving deterministically under combinators.*

4. ***Cross-domain homomorphism.*** *For encodings μ_{D_i} respecting combinator linearity, there exist constructive mappings $h_{i\to j}$ such that $\mu_{D_j}(\star(x, y)) = h_{i\to j}(\mu_{D_i}(\star(x, y)))$.*

5. ***Computational completeness (conditional).*** *If $\mathcal{C}$ contains chunk/shift, conditional branching, and integer arithmetic, then UNNS can simulate finite automata and, with unbounded nesting, Turing-complete models.*

3 Proof Sketches

Part 1. Embed the recurrence by defining a combinator that computes $\tilde{s}_{n+1} = c_1 s_n + \cdots + c_r s_{n-r+1}$, with seeds from $\mathcal{G}$. Induction shows $\mu_{\mathbb{Z}}(s_n) = a_n$.

Part 2. Standard linear algebra: $a_n = \sum_i A_i r_i^n$, dominated by r_d^n. Ratios converge to r_d. Geometric encoding μ_G turns this into a spiral.

Part 3. Since combinators use integer arithmetic, residues evolve deterministically modulo m. Finite-state dynamics follow.

Part 4. Because encodings preserve algebraic structure, the homomorphism $h_{i\to j}$ is explicitly computable.

Part 5. Finite tags and chunk/shift rules can encode finite automata. With unbounded nesting and conditional branching, simulate cyclic tag systems, hence Turing completeness.

4 Corollaries and Examples

Corollary 1 (Fibonacci). *Embedding $F_n = F_{n-1} + F_{n-2}$ produces $\lim_{n\to\infty} F_{n+1}/F_n = \varphi$, yielding the golden ratio spiral under μ_G.*

Corollary 2 (Tribonacci). *Embedding $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ produces limit ratio $\psi \approx 1.839$, yielding a distinct attractor.*

5 Limitations

- Linear recurrences are directly handled; nonlinear cases need separate treatment.
- Homomorphisms require encoding regularity.
- Turing universality requires unbounded nests.
- Numerical demos may suffer precision issues; proofs are exact algebraic.

[11pt]article amsmath,amssymb,amsthm geometry margin=1in
The UNNS Many-Faces Theorem: Formalization and Proof Sketches Research Note
Definition Theorem Lemma Corollary

Abstract

We propose and sketch a formal theorem—the "Many-Faces Theorem"—for the UNNS framework (Unbounded Nested Number Sequences / Universal Network Nexus System). The result shows how linear recurrence sequences, attractors, modular domains, and cross-domain homomorphisms are all naturally embeddable in UNNS. The theorem justifies the phrase "many faces" as algebraic, geometric, topological, number-theoretic, and computational perspectives coexist under a single substrate.

6 Definitions

Definition 6 (Nest Set)**.** *Let S be a set of* nests*, i.e., symbolic tokens or integer values.*

Definition 7 (Combinator Algebra)**.** *Let $\mathcal{C}$ be a finite set of operators (combinators) $\star : S^k \to S$. Each $\star$ is specified by a finite algorithm using integer arithmetic, addition, multiplication, shifting (index arithmetic), and modular arithmetic.*

Definition 8 (Generators and Dynamics)**.** *A UNNS configuration is $U = (S, \mathcal{C}, \mathcal{G})$ with $\mathcal{G} \subset S$ as a finite seed set. An update rule applies combinators iteratively to produce new nests.*

Definition 9 (Domain Mappings)**.** *A* domain *D is any mathematical structure (ring, metric space, manifold, graph). A mapping $\mu_D : S \to D$ is* computable *if an algorithm produces $\mu_D(s)$ from a finite description of s. Typical domains:*

- *A: algebraic (polynomials, vectors)*
- *G: geometric ($\mathbb{R}^2$ polar coordinates)*
- *T: topological (graphs)*
- *$\mathbb{Z}_m$: modular residues*

Definition 10 (UNNS System)**.** *A UNNS system is a tuple $(S, \mathcal{C}, \mathcal{G}, \{\mu_D\}_{D\in\mathcal{D}})$ with a finite set of domains $\mathcal{D}$ and computable domain mappings.*

Assumptions. We assume: (1) each combinator is definable by a finite algorithm with bounded lookback; (2) each μ_D is computable and respects linearity of combinators.

7 The Many-Faces Theorem

Theorem 2 (UNNS Many-Faces Theorem)**.** *Let $U = (S, \mathcal{C}, \mathcal{G}, \{\mu_D\}_{D\in\mathcal{D}})$ be a UNNS system under the above assumptions. Then:*

1. ***Linear recurrence embedding.*** *Any linear recurrence $a_n = c_1 a_{n-1} + \cdots + c_r a_{n-r}$ can be embedded: there exists $s_n \in S$ such that $\mu_{\mathbb{Z}}(s_n) = a_n$.*
2. ***Dominant-root attractor.*** *If the characteristic polynomial has a unique dominant root r_d, then $\lim_{n\to\infty} \mu_{\mathbb{R}}(s_{n+1})/\mu_{\mathbb{R}}(s_n) = r_d$. Geometric embedding μ_G yields logarithmic spirals with growth parameter $\ln |r_d|$.*
3. ***Modular domain partition.*** *For any modulus m, residues $\mu_{\mathbb{Z}_m}(s_n)$ partition nests into m domains, evolving deterministically under combinators.*
4. ***Cross-domain homomorphism.*** *For encodings μ_{D_i} respecting combinator linearity, there exist constructive mappings $h_{i\to j}$ such that $\mu_{D_j}(\star(x,y)) = h_{i\to j}(\mu_{D_i}(\star(x,y)))$.*
5. ***Computational completeness (conditional).*** *If $\mathcal{C}$ contains chunk/shift, conditional branching, and integer arithmetic, then UNNS can simulate finite automata and, with unbounded nesting, Turing-complete models.*

8 Proof Sketches

Part 1. Embed the recurrence by defining a combinator that computes $\tilde{s}_{n+1} = c_1 s_n + \cdots + c_r s_{n-r+1}$, with seeds from $\mathcal{G}$. Induction shows $\mu_{\mathbb{Z}}(s_n) = a_n$.

Part 2. Standard linear algebra: $a_n = \sum_i A_i r_i^n$, dominated by r_d^n. Ratios converge to r_d. Geometric encoding μ_G turns this into a spiral.

Part 3. Since combinators use integer arithmetic, residues evolve deterministically modulo m. Finite-state dynamics follow.

Part 4. Because encodings preserve algebraic structure, the homomorphism $h_{i\to j}$ is explicitly computable.

Part 5. Finite tags and chunk/shift rules can encode finite automata. With unbounded nesting and conditional branching, simulate cyclic tag systems, hence Turing completeness.

9 Corollaries and Examples

Corollary 3 (Fibonacci)**.** *Embedding $F_n = F_{n-1} + F_{n-2}$ produces $\lim_{n\to\infty} F_{n+1}/F_n = \varphi$, yielding the golden ratio spiral under μ_G.*

Corollary 4 (Tribonacci)**.** *Embedding $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ produces limit ratio $\psi \approx 1.839$, yielding a distinct attractor.*

10 Limitations

- Linear recurrences are directly handled; nonlinear cases need separate treatment.
- Homomorphisms require encoding regularity.
- Turing universality requires unbounded nests.
- Numerical demos may suffer precision issues; proofs are exact algebraic.

11 Experimental Roadmap

1. Implement recurrence embedding and validate with standard sequences.
2. Visualize attractors in polar coordinates.
3. Color sequences by residues to reveal modular attractors.
4. Demonstrate homomorphisms via algebra-to-geometry mapping.
5. Explore universality by simulating tag systems.