UNNS Operators: From Grammar to Implementation

UNNS Research Group

Abstract

We present a systematic study of UNNS operators, the elementary transformations of the recursive substrate. Building upon the operational grammar of Inletting, Inlaying, Collapse, Evaluation, Adoption, and Normalization, we formalize operators, classify their roles, and provide mappings to computational implementations as used in the *Dual Sequence Explorer*. Worked examples demonstrate the dynamics of classical sequences under UNNS operators, highlighting divergence, stabilization, and resonance. This paper serves both as a reference manual and as a bridge between abstract UNNS theory and computational practice.

1 Introduction

Operators in the UNNS substrate act as the grammar rules of recursion. They determine how symbolic echoes evolve, stabilize, collapse, or propagate outward. Just as in physics fields are shaped by symmetries, in UNNS sequences are shaped by operators. This paper introduces the operator framework, links it to computational practice, and illustrates its significance with examples.

2 Formal Definitions

Definition 1 (Operator)**.** *A* UNNS operator *is a map*

$$\mathcal{O} : \mathbb{C} \to \mathbb{C},$$

acting on recursive states, which modifies their evolution according to the substrate grammar. Operators may be linear or nonlinear, deterministic or stochastic, but always act within the recursive field.

Definition 2 (Operator Classes)**.** *We classify operators into six main classes:*

1. ***Collapse:*** *Absorbs instabilities into zero or a neutral state.*
2. ***Inlaying:*** *Projects recursion onto a lattice structure.*
3. ***Inletting:*** *Introduces new energy or symbols into the recursion.*
4. ***Adoption:*** *Extends recursion outward, incorporating external states.*
5. ***Evaluation:*** *Fixes recursion into discrete symbolic outputs.*
6. ***Normalization:*** *Balances or damps recursive growth.*

3 Mapping Table

Implemented Operator	UNNS Grammar	Conceptual Role
`threshold_zero`	Collapse	Absorbs small echoes into substrate (zero)
`merge_close`	Collapse/Normalize	Forces near-equal states to converge
`gaussian_round`	Inlaying ($\mathbb{Z}[i]$)	Snap recursion to Gaussian lattice
`eisenstein_round`	Inlaying ($\mathbb{Z}[\omega]$)	Snap recursion to hexagonal lattice
`cyclotomic_proj(`p`)`	Inletting/Inlaying	Embed into pth root of unity lattice
`roundInt / roundFixed`	Evaluation	Snap recursion to discrete numeric states
`damp(`α`)`	Normalize	Dissipates recursive energy, stabilizes growth
`adopt_shift(`k`)`*	Adoption (proposed)	Imports new states from deeper nests

Table 1: Mapping of code-level operators to UNNS substrate grammar. (*) Adoption operator not yet implemented.

4 Worked Examples

4.1 Fibonacci under Collapse

Consider the Fibonacci recursion $F_{n+1} = F_n + F_{n-1}$. Applying the collapse operator $\mathcal{C}(x) = 0$ if $|x| < \epsilon$ eventually erases all small deviations, yielding a purified growth that emphasizes the golden ratio attractor.

4.2 Logistic Map under Inlaying

The logistic recursion $x_{n+1} = rx_n(1 - x_n)$ can be projected onto the Gaussian lattice $\mathbb{Z}[i]$ via `gaussian_round`. The chaotic orbit is then discretized into a 2D lattice walk, revealing structural symmetries hidden in the continuum.

4.3 Prime Gaps under Normalization

Prime gaps $g_n = p_{n+1} - p_n$ are irregular. Applying dampening $\mathcal{N}(x) = \alpha x$ with $0 < \alpha < 1$ exposes underlying resonance frequencies, highlighting a near-periodic echo in the distribution.

5 Geometric Interpretation

Operators can be viewed geometrically:

- Collapse = contraction into the zero node.
- Inlaying = snapping to lattice tilings (square, hexagonal, cyclotomic).

- Adoption = outward spiraling, embedding new shells.
- Normalization = curvature flattening, stabilizing flows.

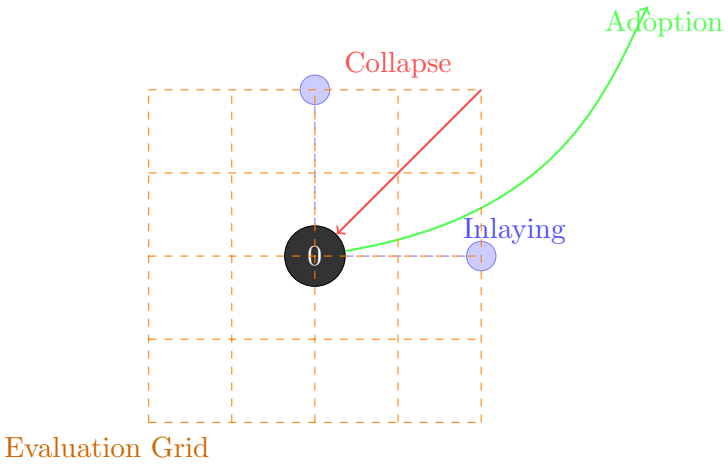

Figure 1: Schematic flow of operators in UNNS geometry.

6 Computational Explorer Reference

The *Dual Sequence Explorer* implements several of these operators in code form, allowing users to compare classical recursion to UNNS recursion side by side. Divergence between the two illustrates the "grammar effect" of the operators.

7 Philosophical Perspective

Operators represent a shift from axiomatic set membership to recursive operational grammar. They are not merely computational tricks, but fundamental rules of symbolic evolution, where mathematics itself is viewed as a substrate of recursive flows.

8 Conclusion

By mapping practical implementations to theoretical operators, we clarify the operational mechanics of UNNS. Future work includes implementing Adoption operators, studying operator composition, and exploring their relation to physical dynamics.