Polynomial Functor Interface Dynamics and Formal State Transition Grammars for Multiversal Boundary Mechanics

Dr. Melvin Sewell, M.Sc., Ph.D.
Bibebibebibe™ Research Initiative | Structured Multiversal Interactions (SMI) Framework

Published via Zenodo Portfolio Records • July 2026

ABSTRACT

This monograph presents a dual-layer mathematical architecture for modeling interaction, rule injection, and state transitions across non-equivalent reality levels within the Structured Multiversal Interactions (SMI) framework [1, 2]. We unify category-theoretic polynomial functor interface dynamics (*Poly*) [3] at the macro-boundary layer with a 4-tuple formal state transition grammar ($G_{TrueFlow}$) [4] at the micro-execution layer. By mapping universe domains as polynomial functors $P_D(y)$, we formalize asymmetric rule projection across Reality Level Scalar (Ξ^{TM}) boundaries [2], guaranteeing structural coherence, non-reciprocal sovereignty, and deterministic state evolution in high-agility multiversal interaction systems [5].

1. Introduction & Structural Problem

Modeling interactions across heterogeneous multiversal domains presents a fundamental mathematical dilemma: how can state transfers occur across systems operating under divergent physical, logic, or narrative laws without causing global state corruption or logical incoherence? Traditional linear state machines and closed-system physical models fail to preserve *sovereignty hierarchy* when high-index systems interact with lower-index sub-domains [1, 5].

To address this, we construct a unified framework operating on two distinct structural tiers:

- **Macro-Interface Layer:** Governed by polynomial functors in the category *Set* [3], handling interface reconfiguration, boundary constraints, and asymmetric rule projection across Reality Level Scalar (Ξ^{TM}) gradients [2].
- **Micro-Execution Layer:** Governed by a formal string rewriting state transition grammar ($G_{TrueFlow}$) [4, 6], executing spatial shifts, energy pulses, and structural locks within permitted state spaces.

2. Category-Theoretic Interface Model (Poly)

Following Spivak et al. [3], we model a universal domain or state space D operating at scalar level Ξ [2] as a polynomial functor $P_D \in Poly$ defined over sets:

$$P_D(y) = \sum_{S \in Positions(D)} y^{Directions(D,\, S)}$$

Where the components are formally specified as:

- $Positions(D)$: The set of attainable, coherent world-states achievable at scalar level Ξ.
- $Directions(D, S)$: The set of admissible input operations, interaction vectors, or directional choices available to external agents when the domain is configured in state S.

2.1 Boundary Transition Morphisms

An interaction between a higher-sovereignty domain D_1 (Ξ_1) and a target domain D_2 (Ξ_2) is modeled as a polynomial functor morphism $\varphi: P_{D_1} \to P_{D_2}$ [3]. The morphism φ consists of two dependent natural mappings:

$$\varphi_{pos}: Positions(D_1) \longrightarrow Positions(D_2)$$

$$\varphi_{dir}(S): Directions(D_2, \varphi_{pos}(S)) \longrightarrow Directions(D_1, S)$$

Sovereignty Law of Asymmetric Rule Injection

When $\Xi_1 > \Xi_2$ [2], the direction-mapping function φ_{dir} is strictly **non-surjective**. The higher-scalar layer D_1 restricts the direction space of D_2, projecting non-reciprocal constraints and enforcing top-down rule dominance while immunizing D_1 from lower-layer state contamination.

3. Formal State Transition Grammar ($G_{TrueFlow}$)

To govern low-level state trajectories, combat movement, and execution vectors within the bounds established by *Poly* [3], we define the formal grammar $G_{TrueFlow}$ as a 4-tuple [4]:

$$G_{TrueFlow} = \langle \Sigma, N, S_0, R \rangle$$

3.1 Grammar Components

Component	Symbol / Set	Structural Definition & Role
Terminal Alphabet	$\Sigma = \{\sigma_{shift}, \sigma_{bind}, \sigma_{pulse}, \sigma_{null}\}$	Primitive atomic operations: dimensional displacement (σ_{shift}), boundary lock (σ_{bind}), scalar change (σ_{pulse}), and identity (σ_{null}).
Non-Terminals	$N = \{S_0, V_{Vector}, B_{Boundary}, T_{Term}\}$	Abstract configuration states representing root state (S_0), trajectory (V_{Vector}), boundary check ($B_{Boundary}$), and equilibrium (T_{Term}).
Axiom State	$S_0 = \langle \Xi_{active}, v, \kappa \rangle$	Initial state vector parameterized by active scalar context (Ξ_{active}), spatial vector (v), and coherence threshold (κ).

3.2 Production Rules (R)

The rewriting rules R specify how valid interaction strings derive [4]:

$$r_1: S_0 \longrightarrow B_{Boundary} \cdot V_{Vector} \qquad \textit{(Initiate state transition)}$$

$$r_2: B_{Boundary} \longrightarrow \sigma_{bind}(\Xi_1, \Xi_2) \qquad \textit{(Evaluate Poly morphism } \varphi \textit{ [3])}$$

$$r_3: V_{Vector} \longrightarrow \sigma_{shift} \cdot V_{Vector} \mid \sigma_{pulse} \cdot T_{Term} \qquad \textit{(Trajectory extension / termination)}$$

$$r_4: T_{Term} \longrightarrow \sigma_{null} \qquad \textit{(Stabilize equilibrium state)}$$

4. Integrated Execution Pipeline

The execution loop guarantees that micro-level moves adhere strictly to macro-level boundary constraints [1, 3]:

STEP 1 • MACRO BOUNDARY QUERY

Poly Interface Verification [3]

An interaction request from domain D_1 (Ξ_1) targeting D_2 (Ξ_2) evaluates $P_{D_1}(y)$ to determine whether state S_1 supports an outgoing transition morphism.

STEP 2 • BOUNDARY RESTRICTION

Morphism Injection & Direction Lock

The morphism φ_{dir} restricts direction space, setting the active interaction scalar context $\Xi_{active} = min(\Xi_1, \Xi_2)$.

STEP 3 • GRAMMAR PARSING

TrueFlow String Rewriting [4, 6]

The grammar parses operations $w \in \Sigma^*$ starting from S_0. Production rule r_2 binds boundary constraints via σ_{bind}.

STEP 4 • TRAJECTORY RESOLUTION

Equilibrium Stabilization

Rule r_3 applies spatial displacements (σ_{shift}) and scalar pulses (σ_{pulse}) until reaching terminal state T_{Term}, collapsing into stable equilibrium via σ_{null}.

5. Theoretical Properties & Proof Sketches

Theorem 1 (Global Soundness): *A state transition sequence is structurally sound if and only if its string derivation* $w \in L(G_{TrueFlow})$ *is accepted by the polynomial boundary morphism* $\varphi: P_{D_1} \to P_{D_2}$ *[3, 4].*

Proof Sketch: Since rule r_2 forces every derivation string to execute σ_{bind} prior to expanding trajectory vectors in r_3, no non-terminal V_{Vector} can evaluate in a state space not previously validated by φ_{dir}. Thus, illegal transitions are unreachable in $L(G_{TrueFlow})$. □

Theorem 2 (Functorial Compositionality): *Sequential multiversal transitions compose naturally under polynomial category laws, preserving sovereignty order [3].*

Proof Sketch: For domains D_1, D_2, D_3, morphisms $\varphi_1: P_{D_1} \to P_{D_2}$ and $\varphi_2: P_{D_2} \to P_{D_3}$ yield a valid composed morphism $(\varphi_2 \,\hat{}\, \varphi_1): P_{D_1} \to P_{D_3}$. Non-surjectivity of φ_{dir} is preserved under composition, ensuring higher-scalar dominance is transitive across arbitrary chain lengths. □

References

[1] **Sewell, M.** (2025). *Structured Multiversal Interactions (SMI): Mapping Narrative and Structural Logic Across Alternate Physical Domains*. Zenodo Repository. DOI: 10.5281/zenodo.sewellsmi.

[2] **Sewell, M.** (2025). *Reality Level Scalar: Ξ™ Protocol Specification*. Bibebibebibe™ Magazine Technical Dossier.

[3] **Spivak, D. I., & Niu, N.** (2022). *Polynomial Functors as a Language for Dynamic Interaction*. Topos Institute Preprint / arXiv:2203.01828.

[4] **Hopcroft, J. E., Motwani, R., & Ullman, J. D.** (2006). *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Addison-Wesley.

[5] **Sewell, M.** (2026). *TrueFlow Universe Architecture and Movement Mechanics Reference Engine*. Zenodo Community Records.

[6] **Mateas, M., & Stern, A.** (2005). *Structuring Content in the Façade Interactive Drama Architecture*. AI Game Engine Programming, 11(3), 211–234.