
FORMALIZATION OF THE EFFECT PROPAGATION PROCESS

Marvin Hansen

marvin.hansen@gmail.com

Draft: August 28, 2025

ABSTRACT

The Effect Propagation Process (EPP), introduced as a novel philosophical framework implemented in the DeepCausality project, fundamentally redefines causality to address the limitations inherent in classical, spacetime-bound conceptions. These limitations become particularly acute when considering insights from contemporary quantum gravity—which suggest emergent spacetime and indefinite causal order—and when tackling the practical complexities of advanced system engineering, such as non-Euclidean data structures, non-linear temporal dynamics, and emergent causal regimes. This paper presents a first-principles mathematical formalization of the EPP, thereby establishing a rigorous and unambiguous foundation crucial for its continued theoretical development, consistent computational instantiation, and broader interdisciplinary engagement.

The core of this formalization lies in precise set-theoretic definitions of EPP’s key components. We define the Contextual Fabric (\mathcal{C}), an explicit and potentially dynamic structure composed of Contextoids and Context Hypergraphs, which can represent diverse Euclidean, non-Euclidean, temporal, and abstract relational information, thus operationalizing EPP’s spacetime agnosticism. Subsequently, we formalize Causal Units (\mathcal{G}), built from operational Causaloids (χ)—unified entities of testable effect transfer that transcend the classical cause-effect dichotomy—and their composition into recursively isomorphic CausaloidGraphs (G) for modeling intricate causal webs. The Causal State Machine (CSM) is then formalized as the mechanism linking causal inferences to deterministic actions.

Beyond these structural entities, the formalism meticulously details the core dynamics of EPP. This includes the Effect Propagation Process (Π_{EPP}) itself, which describes how the activation states of Causaloids evolve through a CausaloidGraph based on their intrinsic functions, contextual inputs, and defined relational logic. Furthermore, we formalize the Operational Generative Function (Φ_{gen}), which provides the mechanisms for the dynamic evolution and emergence of both the Contextual Fabric and the CausaloidGraph structures, a key feature enabling the modeling of emergent causality and regime shifts. The paper also demonstrates how foundational EPP philosophical principles, such as the emergence of temporal order and congruence with classical causality under specific conditions, are naturally reflected within this mathematical framework.

By providing this clear mathematical language, this work aims to enhance the overall clarity, consistency, and testability of the EPP. It seeks to empower researchers and engineers across diverse fields to apply EPP for modeling complex, context-dependent, and dynamically evolving causal phenomena, fostering a more robust and nuanced understanding of causality in both theoretical and practical domains.

Keywords: Computational Causality · Context-Aware Reasoning · Causaloid · Causal Models · Explainable AI · Effect Propagation Process

Contents

1	The Formalization of the Effect Propagation Process	3
1.1	Axioms	3
1.2	Formal Derivation of the EPP Primitives	3
1.3	The PropagatingEffect Primitive (Π_E)	4
1.4	Contextual Fabric (C)	4
1.4.1	The Contextoid (v)	4
1.4.2	The Context Hypergraph (C)	6
1.4.3	Context Accessor (ContextAccessor C_refs)	6
1.5	Causal Units and Structures (G)	6
1.5.1	The Causaloid Primitive (χ)	7
1.5.2	The Causaloid Collection (\mathcal{X}_{coll})	7
1.5.3	The CausaloidGraph (G)	8
1.6	Effect Propagation, Dynamics, and Emergence	9
1.6.1	The Generative Function (Φ_{gen})	10
1.6.2	Core EPP Principles	11
1.7	The Model and Verification Layer	12
1.7.1	Observation, Assumption, and Inference	12
1.7.2	Model (\mathcal{M})	12
1.8	The Causal State Machine (CSM)	12
1.8.1	CSM Operation: State Evaluation and Action Triggering	13
1.8.2	CSM Dynamics: Managing State-Action Pairs	13
1.9	The Teloid and Effect Ethos	14
1.9.1	The Teloid (τ)	14
1.9.2	Proposed Action and Verdict	14
1.9.3	The Effect Ethos Structure	14
1.9.4	The Deontic Inference Process	15
1.10	EPP Example: Smoking, Tar, and Cancer	15
1.10.1	Defining the Causal Problem	15
1.10.2	Formalizing the Components	15
1.10.3	Formalizing Effect Propagation (Π_{EPP})	16
1.10.4	Observations	17
1.11	Discussion	17
2	Conclusion	18
	References	19
	Glossary	19

1 The Formalization of the Effect Propagation Process

1.1 Axioms

The theoretical foundation of the Effect Propagation Process is derived from a single axiom, which is itself a necessary consequence of detaching causality from a presupposed spacetime.

Axiom 1 (of Causality). *Every causal phenomenon is an instance of a spacetime-agnostic functional dependency. Let \mathcal{C} be the universal set of all causal phenomena. The axiom states:*

$$\forall c \in \mathcal{C}, \exists f \text{ such that } c \equiv (E_2 = f(E_1)) \quad (1)$$

where E_1 and E_2 are effects and f is a function. The nature of E and f are deliberately unconstrained.

1.2 Formal Derivation of the EPP Primitives

The core architectural primitives of the Effect Propagation Process are not a collection of independent design choices, but are necessary logical consequences that follow from the single causal axiom. This section provides a formal summary of that derivation.

1. The Contextual Fabric (\mathcal{H}): The Background Condition.

A function $f(E_1)$ may be conditioned on a set of background variables, K . The general form of the axiom is thus $E_2 = f(E_1, K)$. A mechanism to hold and provide this background information K is therefore necessary.

\implies Let the **Contextual Fabric** (\mathcal{H}) be this mechanism.

2. The Causaloid (χ): The Functional Primitive.

Since every causal phenomenon c is defined by a function f , a computational primitive to encapsulate f is a logical necessity.

\implies Let the **Causaloid** (χ) be this primitive, whose primary component is the **causal function** (f_χ).

3. The PropagatingEffect (Π_E): The Isomorphic Message.

A function requires inputs and produces an output. Given the compositional nature of functions (e.g., $g(f(E))$), where the output of one serves as the input to the next, a single, uniform message type is logically necessary.

\implies Let the **PropagatingEffect** (Π_E) be this isomorphic primitive serving as input to and output from a Causaloid.

4. The Causaloid Graph (\mathcal{L}): The Structural Composition.

A system may contain multiple causal phenomena (c_1, c_2, \dots, c_n) that are themselves causally related. A structure is required to represent the set of Causaloids

$\chi_{i_1},$

$\chi_{i_2}, \dots,$

χ_{i_n} and the dependency pathways of their ‘PropagatingEffect’s.

\implies Let the **Causaloid Graph** (\mathcal{L}) be a hypergraph where nodes are Causaloids and hyperedges are propagation effects.

The subsequent sections provide the detailed set-theoretic definitions of these primitives.

1.3 The PropagatingEffect Primitive (Π_E)

The **PropagatingEffect** (Π_E) is the isomorphic message that serves as both input and output for Causaloids. It is a polymorphic type representing the disjoint union of several distinct kinds of effects.

Definition 1.1 (PropagatingEffect). *A PropagatingEffect $\pi \in \Pi_E$ is an element of the set defined by the disjoint union:*

$$\Pi_E = \Pi_{None} \cup \Pi_{Det} \cup \Pi_{Num} \cup \Pi_{Prob} \cup \Pi_{Link} \cup \Pi_{Map} \cup \Pi_{Graph} \cup \Pi_{Relay}$$

where:

- Π_{None} is a singleton set representing the absence of an effect.
- $\Pi_{Det} = \{Deterministic(b) \mid b \in \{true, false\}\}$ represents a boolean effect.
- $\Pi_{Num} = \{Numerical(n) \mid n \in \mathbb{R}\}$ represents a scalar numerical effect.
- $\Pi_{Prob} = \{Probabilistic(p) \mid p \in [0, 1] \subset \mathbb{R}\}$ represents a probability.
- $\Pi_{Link} = \{ContextualLink(id_C, id_v) \mid id_C \in \mathbb{I}, id_v \in \mathbb{I}\}$ is a reference to a Contextoid within a Context.
- $\Pi_{Map} = \{Map(m) \mid m : \mathbb{I} \rightarrow \Pi_E\}$ is a map from identifiers to other PropagatingEffects.
- $\Pi_{Graph} = \{Graph(g) \mid g \text{ is a graph of } \Pi_E\}$ represents a graph of effects.
- $\Pi_{Relay} = \{RelayTo(i, \pi) \mid i \in \mathbb{N}, \pi \in \Pi_E\}$ is a control-flow directive for adaptive reasoning, instructing the engine to jump to the Causaloid at index i with a new input effect π .

1.4 Contextual Fabric (\mathcal{C})

The Effect Propagation Process (EPP), being spacetime-agnostic, requires a formally defined structure through which effects propagate and within which causal relationships are conditioned. This structure is termed the **Contextual Fabric**, denoted abstractly as \mathcal{C} . The Contextual Fabric is not a monolithic entity but can be composed of multiple, distinct contextual realms, each potentially representing different aspects of a system's environment or internal state (e.g., physical space, temporal scales, relational networks, data streams). This section provides a set-theoretic formalization of this fabric, starting from its most granular component, the Contextoid, and building up to collections of Context Hypergraphs.

1.4.1 The Contextoid (v)

The atomic unit of contextual information within the EPP framework is the **Contextoid**. A Contextoid encapsulates a single, identifiable piece of data, which can be temporal, spatial, spatiotemporal, symbolic, or a general data value.

Definition 1.2 (Contextoid). *A Contextoid v is an element of a set of all possible contextoids $V_{\mathcal{C}}$. It is formally defined as a tuple:*

$$v = (id_v, payload_v, adj_v)$$

where $id_v \in \mathbb{I}$ is a unique identifier, $payload_v$ is the content-bearing element, and adj_v is an optional protocol for dynamic modification.

Contextoid Payload

Definition 1.3 (Contextoid Payload). *The payload _{v} represents the actual contextual information. It is an element from the disjoint union of several universal sets, each representing a fundamental category of contextual information:*

$$payload_v \in \mathcal{D} \cup \mathcal{T} \cup \mathcal{S} \cup \mathcal{ST} \cup \Sigma$$

where \mathcal{D} is the set of data payloads, \mathcal{T} is the set of temporal payloads, \mathcal{S} is the set of spatial payloads, \mathcal{ST} is the set of spacetime payloads, and Σ is the set of symbolic payloads.

Data Payload

Definition 1.4 (Data Payloads (\mathcal{D})). *Let \mathcal{D} be the universal set of all data payloads. Each element $d \in \mathcal{D}$ is a tuple $(id_d, data_d)$, where $id_d \in \mathbb{I}$ and $data_d$ is a value from a generic domain \mathcal{V}_D .*

Temporal Payloads

Definition 1.5 (Temporal Payloads (\mathcal{T})). *Let \mathcal{T} be the universal set of all temporal payloads. Each element $t \in \mathcal{T}$ is a tuple $(id_t, scale_t, unit_t)$.*

- $id_t \in \mathbb{I}$ is its unique identifier.
- $scale_t \in \mathcal{T}_{scale}$, where \mathcal{T}_{scale} is a predefined enumerated set of temporal granularities (e.g., $\{Seconds, Ticks, SymbolicEvent\}$).
- $unit_t \in \mathcal{V}_T$, where \mathcal{V}_T is a suitable ordered value set (e.g., \mathbb{R} or \mathbb{N}).

The set \mathcal{T} may be partitioned into subsets corresponding to different temporal semantics, such as discrete, continuous, or symbolic time.

Spatial Payloads

Definition 1.6 (Spatial Payloads (\mathcal{S})). *Let \mathcal{S} be the universal set of all spatial payloads. Each element $s \in \mathcal{S}$ is defined by a tuple $(id_s, coords_s)$.*

- $id_s \in \mathbb{I}$ is its unique identifier.
- $coords_s$ is a vector (c_1, c_2, \dots, c_n) where each component $c_i \in \mathcal{V}_S$ for some value set \mathcal{V}_S .
- The dimensionality n is given by a function $f_{dim} : \mathcal{S} \rightarrow \mathbb{N}$.

For any subset of \mathcal{S} , a metric $d : (\mathcal{S}, \mathcal{S}) \rightarrow \mathbb{R}^+$ may be defined, allowing for distance calculations. This structure accommodates both Euclidean and non-Euclidean geometries (e.g., geodetic, rotational).

Spacetime Payloads

Definition 1.7 (Spacetime Payloads (\mathcal{ST})). *Let \mathcal{ST} be the universal set of all spacetime payloads. Each element $st \in \mathcal{ST}$ is a composite structure, formally a pair (s, t) , where $s \in \mathcal{S}$ and $t \in \mathcal{T}$.*

Symbolic Payloads

Definition 1.8 (Symbolic Payloads (Σ)). *Let Σ be the universal set of all symbolic payloads. Each element $sym \in \Sigma$ is a tuple $(id_{sym}, repr_{sym})$, where $id_{sym} \in \mathbb{I}$ and $repr_{sym}$ is an element from a domain of symbolic representations \mathcal{R}_{sym} .*

This polymorphic design allows the EPP to model and reason about highly heterogeneous and complex contextual information in a consistent and formally grounded manner, closing the gap between abstract theory and practical implementation.

Contextoid Adjustability Protocol

The component adj_v represents optional functions implementing the EPP's Adjustable protocol, allowing for the dynamic modification of a Contextoid's payload $payload_v$. Let $\mathcal{V}_{payload}$ be the set of all possible payload values and \mathcal{V}_{adj_factor} be the set of all possible adjustment factors. The functions are:

- $update_v : \mathcal{V}_{payload} \rightarrow \text{void}$: This function replaces the current $payload_v$ of the Contextoid with a new payload value.
- $adjust_v : \mathcal{V}_{adj_factor} \rightarrow \text{void}$: This function modifies the current $payload_v$ based on an adjustment factor.

The 'void' return type indicates that these functions perform a stateful update on the Contextoid v . Their absence implies the Contextoid's payload is immutable post-instantiation.

1.4.2 The Context Hypergraph (C)

Individual Contextoids are organized into a structured collection called a **Context Hypergraph**. This hypergraph structure allows for the representation of complex, N-ary relationships between different pieces of contextual information, which is crucial for modeling the intricate dependencies found in real-world systems.

Definition 1.9 (Context Hypergraph). *An individual Context Hypergraph C is defined as a tuple:*

$$C = (V_C, E_C, ID_C, Name_C)$$

where V_C is its finite set of **Contextoid** nodes (each conforming to Definition 3.1, thus $V_C = \{v_1, v_2, \dots, v_n\}$), E_C is its finite set of **Hyperedges**, $ID_C \in \mathbb{N}$ is a unique identifier for C , and $Name_C$ is a descriptive name.

Set of Context Hyperedges (E_C). A Hyperedge $e \in E_C$ represents a relationship among a subset of Contextoids in V_C . It is defined as a tuple:

$$e = (V_e, \text{kind}_e, \text{label}_e)$$

where:

- $V_e \subseteq V_C$ is a non-empty subset of Contextoids in C connected by this hyperedge, with $|V_e| \geq 1$.
- $\text{kind}_e \in \mathcal{K}_{\text{relation}}$ specifies the type or nature of the relationship, drawn from a predefined set of possible relation kinds $\mathcal{K}_{\text{relation}}$ (e.g., $\mathcal{K}_{\text{relation}} = \{\text{containment}, \text{proximity}, \text{temporal_sequence}, \text{logical_and}, \text{synonymy}\}$).
- label_e is an optional descriptive label for the hyperedge (e.g., a string).

The use of hyperedges allows a single relational link (e) to connect an arbitrary number of Contextoids (V_e), enabling the representation of multi-way contextual dependencies.

1.4.3 Context Accessor ($\text{ContextAccessor}(\mathcal{C}_{\text{refs}})$)

To enable Causaloids (defined in Section 1.5) to interact with the Contextual Fabric, a mechanism for querying and retrieving contextual information is required. This is conceptualized as a **Context Accessor**.

Definition 1.10 (Context Accessor). *A Context Accessor, denoted $\text{ContextAccessor}(\mathcal{C}_{\text{refs}})$, is a functional interface that provides read-access to a specified subset of Context Hypergraphs $\mathcal{C}_{\text{refs}} \subseteq \mathcal{C}_{\text{sys}}$.*

Its operations, denoted abstractly, would typically include functions f_{access} such that:

- $\text{getContextoid}(id_v, ID_C) \rightarrow v \cup \{\text{null}\}$: Retrieves a Contextoid v by its identifier id_v from a specific Context Hypergraph C (identified by ID_C) within $\mathcal{C}_{\text{refs}}$.
- $\text{getHyperedges}(v_i, \text{kind}_e) \rightarrow \{e_1, \dots, e_m\}$: Retrieves all hyperedges of a specific kind kind_e that involve a given Contextoid v_i .

The precise set of functions within the Context Accessor depends on the requirements of the Causaloid functions that utilize it. This formalism defines the accessor as a means to obtain relevant contextual data for causal evaluation based on identifiers and relational queries.

1.5 Causal Units and Structures (\mathcal{G})

Having formalized the Contextual Fabric (\mathcal{C}) within which effects propagate, we now turn to the formalization of the causal entities themselves. The Effect Propagation Process (EPP) posits that causal influence is mediated by discrete, operational units which can be composed into complex structures. This section defines these Causal Units and Structures, denoted abstractly as \mathcal{G} . The central entity is the Causaloid (χ), representing an individual causal mechanism. These primitives are then composed into Causaloid Collections and CausaloidGraphs (G) to model intricate webs of causal relationships.

1.5.1 The Causaloid Primitive (χ)

The fundamental unit of causal interaction within EPP is the **Causaloid**. It is a recursively isomorphic primitive that encapsulates a single, testable causal mechanism.

Definition 1.11 (Causaloid). *A Causaloid χ is a tuple:*

$$\chi = (id_\chi, type_\chi, f_\chi, \mathcal{C}_{refs}, desc_\chi, \mathcal{A}_{linked}, I_{linked})$$

where:

- $id_\chi \in \mathbb{I}$ is a unique identifier.
- $type_\chi \in \{\text{Singleton}, \text{Collection}, \text{Graph}\}$ specifies its structural nature.
- f_χ is the **causal function** associated with the Causaloid. Its precise signature depends on $type_\chi$.
- $\mathcal{C}_{refs} \subseteq \mathcal{C}_{sys}$ is a set of references to Context Hypergraphs for the evaluation of f_χ .
- $desc_\chi, \mathcal{A}_{linked}, I_{linked}$ are optional metadata for description and empirical grounding.

Causal Function for Singletons ($f_{\chi,S}$) For a Causaloid of type Singleton, the causal function is a direct evaluation mapping an input effect and a context to an output effect:

$$f_{\chi,S} : (\Pi_E \times \text{ContextAccessor}(\mathcal{C}_{refs})) \rightarrow \Pi_E$$

1.5.2 The Causaloid Collection (\mathcal{X}_{coll})

A **Causaloid Collection** models causal phenomena that arise from the aggregation of multiple individual causes. It encapsulates a set of Causaloids and an explicit logic for combining their effects. This structure is fundamental for representing many-to-one causal relationships in a concise and manageable way.

Definition 1.12 (Causaloid Collection). *A Causaloid Collection, \mathcal{X}_{coll} , is a set of Causaloids:*

$$\mathcal{X}_{coll} = \{\chi_1, \chi_2, \dots, \chi_p\}$$

Aggregate Logic (\mathcal{A}_L) The evaluation of a collection is governed by a specific aggregation rule, which determines how the effects of the individual member Causaloids are combined into a single, coherent outcome.

Definition 1.13 (Aggregate Logic). *Let \mathcal{A}_L be the enumerated set of aggregate logics:*

$$\mathcal{A}_L = \{\text{All}, \text{Any}, \text{None}, \text{Some}(k)\} \text{ where } k \in \mathbb{N}$$

Each logic corresponds to a standard logical operator applied to the set of evaluated effects from the collection's members:

- **All**: Logical conjunction (\wedge). The collection is active only if all member Causaloids are active.
- **Any**: Logical disjunction (\vee). The collection is active if at least one member Causaloid is active.
- **None**: Negated disjunction ($\neg\vee$). The collection is active only if no member Causaloids are active.
- **Some(k)**: k -out-of- n threshold logic. The collection is active if at least k member Causaloids are active.

Causal Function for Collections ($f_{\chi,C}$) For a Causaloid of type Collection, the causal function evaluates each member of its collection and aggregates the results using a specified logic.

Definition 1.14 (Causal Function for Collections). *The function is a mapping:*

$$f_{\chi,C} : (\mathcal{P}(\chi) \times \Pi_E \times \mathcal{A}_L) \rightarrow \Pi_E$$

This function, $f_{\chi,C}$, takes a set of causaloids (the collection \mathcal{X}_{coll}), an input effect $\pi_{in} \in \Pi_E$, and an aggregate logic $l \in \mathcal{A}_L$. It computes a set of intermediate effects $\{\pi_1, \pi_2, \dots, \pi_p\}$ by applying π_{in} to each $\chi_i \in \mathcal{X}_{coll}$, and then applies the logic l to this set to produce a single, final output effect $\pi_{out} \in \Pi_E$.

1.5.3 The CausaloidGraph (G)

Individual Causaloids are composed into **CausaloidGraphs** to represent complex webs of interconnected causal relationships. A CausaloidGraph is itself a hypergraph.

Definition 1.15 (CausaloidGraph). A CausaloidGraph G is defined as a tuple:

$$G = (V_G, E_G, ID_G, Name_G)$$

where:

- V_G is a finite set of causal nodes.
- E_G is a finite set of causal hyperedges, representing functional relationships or propagation pathways between nodes in V_G .
- $ID_G \in \mathbb{I}$ (or \mathbb{N}) is a unique identifier for the CausaloidGraph.
- $Name_G$ is a descriptive name for the CausaloidGraph (e.g., a string).

Definition 1.16 (Set of Causal Nodes (V_G)). Each causal node $v_g \in V_G$ is defined as a tuple $v_g = (id_g, payload_g)$, where $id_g \in \mathbb{I}$ is its unique identifier within G . The payload _{g} embodies the principle of **recursive isomorphism** central to EPP, allowing for hierarchical model construction. It can be one of:

- A single Causaloid χ (as per Definition 4.1).
- A collection of Causaloids $\{\chi_1, \chi_2, \dots, \chi_m\}$, where the collection itself might have aggregate evaluation logic (e.g., "active if any χ_i is active").
- Another entire CausaloidGraph G' , enabling the nesting of causal sub-models.

Definition 1.17 (Set of Causal Hyperedges). Each causal hyperedge $e_g \in E_G$ represents a directed functional relationship or pathway for effect propagation. It is defined as a tuple:

$$e_g = (V_{source}, V_{target}, logic_e)$$

where:

- $V_{source} \subseteq V_G$ is a non-empty set of source causal nodes.
- $V_{target} \subseteq V_G$ is a non-empty set of target causal nodes.
- $logic_e$ defines the functional relationship or condition under which effects propagate from V_{source} to V_{target} . This logic might range from simple conjunction/disjunction of source node states to more complex functions that determine how the activation of source nodes influences target nodes.

This hypergraph structure allows for many-to-many causal relationships between causaloids.

Definition 1.18 (State of the CausaloidGraph (S_G)). The state of a CausaloidGraph G at any given point in its evaluation is defined by the activation states of its constituent causal nodes.

Definition 4.3 (State of CausaloidGraph): The state S_G is a function mapping each causal node in V_G to an activation status:

$$S_G : V_G \rightarrow \{\text{active}, \text{inactive}\}$$

This state evolves as effects propagate through the graph according to the Causaloid functions and the logic of the hyperedges.

1.6 Effect Propagation, Dynamics, and Emergence

The preceding sections formalized the static structural components of the Contextual Fabric (\mathcal{C}) and Causal Units/Structures (\mathcal{G}). This section now delves into the heart of the Effect Propagation Process (EPP): its core dynamics, the mechanisms for generating and evolving these structures, and how these formalisms embody the foundational philosophical principles of EPP. This operationalization is crucial for understanding how EPP moves beyond classical causality to address complex, dynamic, and emergent systems.

The Effect Propagation Process (Pi_EPP)

The core dynamic of EPP is the propagation of effects through a CausaloidGraph G , influenced by the Contextual Fabric \mathcal{C}_{sys} and triggered by observations or internal states. This process, denoted Π_{EPP} , describes how the activation state S_G of the CausaloidGraph evolves from an initial state S_G to an updated state S'_G .

Definition 1.19 (Defining an Effect (epsilon) within EPP). *Within the EPP formalism, an "Effect" (ϵ) is primarily represented by:*

1. *The activation state (active or inactive) of a Causaloid (χ) or a causal node (v_g) within a CausaloidGraph, as determined by its causal function f_χ .*
2. *The transfer of this activation status, or information derived from it, to other connected Causaloids according to the graph structure and hyperedge logic.*

An active state $S_G(v_g) = \text{active}$ signifies that the causal mechanism encapsulated by v_g has met its conditions for effect transfer.

The nature of the effect is qualitatively described by desc_χ , and its consequence is how its activation influences other parts of the CausaloidGraph.

Definition 1.20 (Input Observations and Triggers (O_input)). *The propagation process Π_{EPP} is typically initiated or influenced by external inputs or observations.*

Definition 5.1 (Input Trigger Set): *Let \mathcal{O}_{input} be the set of all possible input observations relevant to the Causaloid-Graph G . An input $o_{in} \in \mathcal{O}_{input}$ is an element compatible with the \mathcal{O}_{type} expected by one or more Causaloid functions f_χ within G . At any given evaluation cycle, a set of specific input observations $\mathcal{O}_{trig} \subseteq \mathcal{O}_{input}$ serves as triggers. These triggers might be:*

- *Directed at specific Causaloids in G (e.g., root nodes, or nodes representing sensors).*
- *Representing updates to the Context Collection \mathcal{C}_{sys} which, in turn, affect the evaluation of context-aware Causaloids.*

Definition 1.21 (The Propagation Transition Function)). *The propagation of effects within a CausaloidGraph G is formalized as a state transition function Π_{EPP} . This function describes how the graph's state evolves. **Definition 5.2 (EPP Transition Function):** The EPP Transition Function Π_{EPP} is a mapping:*

$$\Pi_{EPP} : (G, S_G, \mathcal{C}_{sys}, \mathcal{O}_{trig}) \rightarrow S'_G$$

where $G = (V_G, E_G, \dots)$ is a CausaloidGraph, $S_G : V_G \rightarrow \{\text{active}, \text{inactive}\}$ is its current state, \mathcal{C}_{sys} is the Context Collection, \mathcal{O}_{trig} is the set of current input triggers, and $S'_G : V_G \rightarrow \{\text{active}, \text{inactive}\}$ is the updated state of the CausaloidGraph.

The computation of S'_G from S_G is achieved through an iterative evaluation process driven by the structure of G and the logic of its components):

1. **Initialization:** *Some nodes in V_G may have their states in S_G initially set or updated directly by \mathcal{O}_{trig} .*
2. **Iterative Evaluation & Propagation:** *The process iteratively (or recursively, depending on the traversal strategy) considers nodes $v_g \in V_G$:*
 - (a) *The activation of a node v_g (i.e., $S'_G(v_g)$) is determined by evaluating its causal function $f_{\chi_{v_g}}$. This evaluation takes into account:*
 - *Relevant observations from \mathcal{O}_{trig} or the activation states of its source nodes (elements of V_{source} from incoming hyperedges e_g).*
 - *Contextual information obtained via $\text{ContextAccessor}(\mathcal{C}_{ref_{\chi_{v_g}}})$.*

- (b) *The influence of source nodes V_{source} on target nodes V_{target} is mediated by the logic_e of the connecting hyperedge $e_g = (V_{source}, V_{target}, logic_e)$. This logic_e determines if and how the states of source nodes contribute to triggering the evaluation of target nodes.*
3. **Convergence:** *The process continues until no further changes in node states occur for the given O_{trig} and C_{sys} , resulting in the stable updated state S'_G . For CausaloidGraphs with cycles, specific convergence criteria or iteration limits may be necessary.*

This operational definition emphasizes that effect propagation is a structured traversal and evaluation, where the "rules" of propagation are embedded in both the individual Causaloid functions f_χ and the connective logic logic_e of the CausaloidGraph's hyperedges.

1.6.1 The Generative Function (Φ_{gen})

The EPP philosophy posits that the Contextual Fabric and even the Causal Structures themselves can be dynamic and emergent, shaped by an underlying "generative function." This formalism operationalizes this concept as Φ_{gen} , a function or set of functions that govern the evolution of C_{sys} and/or G .

Conceptual Role of Phi_gen in EPP

Φ_{gen} represents the meta-level rules or processes that can alter the EPP's structural components. It embodies the system's capacity for adaptation, learning, or emergence beyond simple state changes within a fixed structure. While the EPP framework itself does not mandate a specific physical interpretation for an ultimate generative function, it provides the means to model systems where such generative dynamics are at play operationally.

Generation and Dynamics of Contexts (Phi_gen_C)

Let Φ_{gen_C} be the component of Φ_{gen} responsible for the evolution of the Context Collection.

Definition 1.22 (Context Generative Function).

$$\Phi_{gen_C} : (C_{sys}, O_{in}, E_{ext}) \rightarrow C'_{sys}$$

where O_{in} represents relevant system inputs/observations and E_{ext} represents external events or triggers. C'_{sys} is the updated Context Collection. This function can:

- *Modify Contextoids $v \in V_C$ within a $C_i \in C_{sys}$ via their adj_v protocol.*
- *Modify the structure of a Context Hypergraph C_i , e.g., by adding/removing Contextoids from V_{C_i} or Hyperedges from E_{C_i} .*
- *Add new Context Hypergraphs to C_{sys} or remove existing ones.*

The specific rules defining Φ_{gen_C} are domain-dependent but allow for contexts that structurally adapt to new information or changing environmental conditions (e.g., the dynamic temporal hypergraph described in the EPP philosophy [1]).

Generation and Dynamics of Causal Structures (Phi_gen_G)

Let Φ_{gen_G} be the component of Φ_{gen} responsible for the evolution of CausaloidGraphs, enabling emergent causality.

Definition 1.23 (Causal Structure Generative Function).

$$\Phi_{gen_G} : (G, S_G, C_{sys}, O_{in}, E_{ext}) \rightarrow (G', S'_G)$$

This function can modify the CausaloidGraph $G = (V_G, E_G, \dots)$ into a new graph $G' = (V'_G, E'_G, \dots)$ with a corresponding initial state S'_G . Modifications can include:

- *Adding or removing causal nodes v_g from V_G .*
- *Modifying the payload of existing nodes (e.g., changing a Causaloid's function f_χ or its linked assumptions \mathcal{A}_{linked}).*
- *Adding or removing causal hyperedges e_g from E_G , or altering their logic_e.*

This function formalizes the EPP's capacity for causal emergence, where the causal "rules of the game" themselves evolve in response to the system's experience and context.

Co-evolution of Context and Causal Structure (Phi_gen_Total)]

Definition 1.24 (Total Generative Function).

$$\Phi_{gen_Total} : (G, S_G, C_{sys}, O_{in}, E_{ext}) \rightarrow (G', S'_G, C'_{sys})$$

This represents the combined action of Φ_{gen_C} and Φ_{gen_G} , allowing for feedback between structural changes in context and structural changes in causal models.

1.6.2 Core EPP Principles

The formalism presented supports and makes explicit the core philosophical principles of EPP.

Spacetime Agnosticism

The EPP formalism is inherently spacetime-agnostic. Neither the Contextoid (Def 1.2), Context Hypergraph (Def 1.9), Causaloid (Def 1.11), nor CausaloidGraph (Def 1.15) definitions mandate a spatiotemporal nature for the underlying fabric or relations. Spatial or temporal properties are introduced only if specific payload_v types (Time, Space, Space-Time) are used within Contextoids. The core propagation function Π_{EPP} (Def 1.1) and generative function Φ_{gen} (Def 1.22) operate on these abstract structures; they only interact with spatiotemporal concepts if explicitly encoded within particular Causaloid functions f_χ or the rules of Φ_{gen} . This formally detaches EPP causality from a presupposed spacetime, fulfilling a key requirement for modeling non-Euclidean systems or systems where spacetime is emergent.

Emergence of Temporal Order

While EPP does not assume a universal linear temporal order, an operational or effective temporal order can emerge within the system through several mechanisms:

- **Contextual Sequencing:** Sequences of Time(t) Contextoids, or relations ($kind_e$) like "temporally_precedes" defined in E_C , can establish a local or domain-specific temporal order within C_{sys} .
- **Propagation Dynamics:** The iterative application of Π_{EPP} inherently defines evaluation steps. While these steps are not necessarily universal time, they form a sequence of state transitions $S_G \rightarrow S'_G \rightarrow S''_G \dots$ that constitutes an internal process time.
- **Generative Function Dynamics:** Changes orchestrated by Φ_{gen} also occur in sequence, creating a history of structural evolution.

Causaloid functions f_χ can then be designed to query and utilize this emergent temporal information from the context or process history when relevant.

Congruence with Classical Causality

Classical causality (A causes B if A precedes B and influences B) can be shown as a special case within the EPP formalism under specific conditions:

1. The Contextual Fabric C_{sys} is structured to represent a classical (possibly dynamic, as in GR) spacetime with a well-defined linear temporal order.
2. Causaloids χ_A and χ_B represent events A and B.
3. A causal hyperedge e_g connects χ_A to χ_B , with $logic_e$ such that χ_B is activated if χ_A is active AND the context (queried via f_{χ_B} or through $logic_e$) confirms that the event corresponding to χ_A occurred at an earlier time than the event corresponding to χ_B within the defined spacetime context.
4. The Causaloid χ_A can be designated "the cause" and χ_B "the effect" based on this emergent, contextually-defined temporal precedence.

Under these (and potentially other simplifying) assumptions, EPP's effect propagation aligns with classical cause-and-effect chains.

1.7 The Model and Verification Layer

The EPP primitives are composed into a verifiable **Model**, which encapsulates the core reasoning logic along with its operational assumptions.

1.7.1 Observation, Assumption, and Inference

These are empirical grounding primitives for a model.

- An **Observation** O is a tuple $(id_O, val_O, effect_O)$ representing an empirical measurement.
- An **Assumption** A is a tuple $(id_A, desc_A, f_A)$ where $f_A : \mathcal{P}(\Pi_E) \rightarrow \{\text{true}, \text{false}\}$ is a verifiable predicate about the operational conditions.
- An **Inference** I is a tuple representing a tested hypothesis about a causal link.

1.7.2 Model (\mathcal{M})

Definition 1.25 (Model). *A Model \mathcal{M} is a tuple:*

$$\mathcal{M} = (id_M, author_M, desc_M, \mathcal{A}_{set}, \chi_M, \mathcal{C}_M)$$

where $id_M, author_M, desc_M$ are metadata, \mathcal{A}_{set} is a set of Assumptions upon which the model's validity depends, χ_M is the root Causaloid representing the model's core logic, and \mathcal{C}_M is the Context it operates on.

1.8 The Causal State Machine (CSM)

The Effect Propagation Process (EPP) framework provides mechanisms for modeling how effects propagate and causal structures evolve. To translate the insights derived from such processes into concrete operations or interventions, the **Causal State Machine (CSM)** is introduced. The CSM serves as an orchestrator, linking specific causal conditions (represented as Causal States) to predefined deterministic Actions.

Let \mathbb{I}_{CSM} be a set of unique identifiers for Causal States within a CSM.

Definition 1.26 (Causal State). *A Causal State q represents a specific condition whose truthfulness is determined by the evaluation of an associated Causaloid. It is defined as a tuple:*

$$q = (id_q, data_q, \chi_q, version_q)$$

where:

- $id_q \in \mathbb{I}_{CSM}$ is a unique identifier for this Causal State.
- $data_q \in \mathcal{O}_{type}$ is the specific input observation or data value (compatible with the Causaloid's expected observation type \mathcal{O}_{type}) used for evaluating this state. This data may be intrinsic to the state or provided externally during evaluation.
- χ_q is a reference to a Causaloid (as defined in Section 1.5.1), $\chi_q = (id_{\chi_q}, f_{\chi_q}, \mathcal{C}_{refs_q}, \dots)$. The function f_{χ_q} embodies the predicate for this state.
- $version_q \in \mathbb{N}$ is an optional version number for the state definition.

The activation of Causal State q is determined by $eval(q) = f_{\chi_q}(data_q, \text{ContextAccessor}(\mathcal{C}_{refs_q}))$.

Definition 1.27 (Causal Action). *A Causal Action a represents a deterministic operation. It is defined as a tuple:*

$$a = (exec_a, descr_a, version_a)$$

where:

- $exec_a : \text{void} \rightarrow \text{Result}(\text{void}, \text{ActionError})$ is the executable function representing the action. Its invocation may modify a WorldState implicitly.
- $descr_a$ is a descriptive label for the action.
- $version_a \in \mathbb{N}$ is an optional version number for the action definition.

Definition 1.28 (Causal State Machine). *A Causal State Machine M is defined by its collection of Causal State-Action pairs. Let \mathcal{Q} be the set of all possible Causal States and \mathcal{A} be the set of all possible Causal Actions.*

$$M = (\mathcal{SA}_M)$$

where:

- $\mathcal{SA}_M \subseteq \{(q, a) \mid q \in \mathcal{Q}, a \in \mathcal{A}\}$ is a finite set of ordered pairs, where each pair (q_j, a_j) associates a unique Causal State q_j (identified by id_{q_j}) with a Causal Action a_j . In implementation, this is often represented as a map from $id_q \rightarrow (q, a)$.

The set of CausaloidGraphs \mathcal{G}_M supervised by M is implicitly defined as the set of all CausaloidGraphs containing any Causaloid χ_q referenced by any q in \mathcal{SA}_M . Similarly, the set of relevant Context Hypergraphs \mathcal{C}_M is the union of all \mathcal{C}_{refs_q} for all q in \mathcal{SA}_M .

1.8.1 CSM Operation: State Evaluation and Action Triggering

The operation of the CSM involves evaluating its Causal States and triggering associated Causal Actions.

Single State Evaluation and Action (eval_single_state): Given a CSM $M = (\mathcal{SA}_M)$, an identifier $id_q^* \in \mathbb{I}_{CSM}$, and potentially new input data $d_{in} \in \mathcal{O}_{type}$ for that state:

1. Retrieve the state-action pair (q^*, a^*) from \mathcal{SA}_M such that $id_{q^*} = id_q^*$. If no such pair exists, an error is indicated.
2. The Causal State $q^* = (id_{q^*}, data_{q^*}, \chi_{q^*}, \dots)$ is evaluated. The evaluation function is $f_{\chi_{q^*}}$. The input data used is either d_{in} (if provided externally for this evaluation) or the state's intrinsic data $data_{q^*}$. Let $trigger = f_{\chi_{q^*}}(data_{eval}, ContextAccessor(\mathcal{C}_{refs_{q^*}}))$.
3. If $trigger = true$, then the action a^* is executed by invoking its function $exec_{a^*}()$.

All States Evaluation and Action (eval_all_states): Given a CSM $M = (\mathcal{SA}_M)$:

1. For each state-action pair $(q_j, a_j) \in \mathcal{SA}_M$:
 - (a) The Causal State $q_j = (id_{q_j}, data_{q_j}, \chi_{q_j}, \dots)$ is evaluated using its intrinsic data $data_{q_j}$. Let $trigger_j = f_{\chi_{q_j}}(data_{q_j}, ContextAccessor(\mathcal{C}_{refs_{q_j}}))$.
 - (b) If $trigger_j = true$, then the action a_j is executed by invoking its function $exec_{a_j}()$.

The order and concurrency of action execution in step 1(b) depend on the CSM's specific execution semantics (e.g., sequential, parallel, prioritized), which are an implementation detail beyond this core formalism.

1.8.2 CSM Dynamics: Managing State-Action Pairs

The set of state-action pairs \mathcal{SA}_M within a CSM M can be dynamic, allowing the CSM to adapt its stimulus-response behavior. Formal operations on \mathcal{SA}_M include:

- **AddStateAction**(M, q_{new}, a_{new}): $\mathcal{SA}'_M = \mathcal{SA}_M \cup \{(q_{new}, a_{new})\}$, provided $id_{q_{new}}$ is not already a key in \mathcal{SA}_M .
- **RemoveStateAction**(M, id_q): $\mathcal{SA}'_M = \mathcal{SA}_M \setminus \{(q, a) \mid id_q \text{ is the identifier of } q\}$.
- **UpdateStateAction**($M, id_q, q_{updated}, a_{updated}$): Replaces the pair (q, a) associated with id_q with $(q_{updated}, a_{updated})$.

1.9 The Teleological Fabric (\mathbb{E})

To ensure that a system's actions remain aligned with pre-defined goals, safety protocols, or ethical principles, the EPP introduces a teleological layer of governance. This is formalized as the **Effect Ethos**, a deontic reasoning engine that evaluates proposed actions from the Causal State Machine against a set of computable norms. This formalism is derived from the Defeasible Deontic Inheritance Calculus (DDIC) [2] and the DeepCausality reference implementation.

1.9.1 The Teloid (τ)

The fundamental, atomic unit of a computable norm is the **Teloid**. It encapsulates a single, verifiable rule.

Definition 1.29 (Teloid). A Teloid τ is defined as a tuple:

$$\tau = (id_\tau, a_{id}, p_\tau, m_\tau, t_\tau, s_\tau, prio_\tau, tags_\tau, meta_\tau)$$

where:

- $id_\tau \in \mathbb{I}$ is a unique identifier for the Teloid.
- a_{id} is a string identifier for the class of action the norm governs.
- $p_\tau : (C_{sys}, A_p) \rightarrow \{true, false\}$ is the **activation predicate**, a function that determines if the norm is active based on the system Context and a Proposed Action.
- $m_\tau \in \{Impermissible, Obligatory, Optional(c)\}$ is the **deontic modality**, where c is an associated integer cost for optional norms.
- $t_\tau \in \mathbb{N}$ is a timestamp for the Lex Posterior heuristic.
- $s_\tau \in \mathbb{N}$ is a specificity value for the Lex Specialis heuristic.
- $prio_\tau \in \mathbb{N}$ is a priority value for the Lex Superior heuristic.
- $tags_\tau$ is a set of string tags for filtering and indexing.
- $meta_\tau$ is an optional set of key-value metadata.

1.9.2 Proposed Action (A_p) and Verdict (V)

The Effect Ethos evaluates a **Proposed Action** and returns a **Verdict**.

Definition 1.30 (Proposed Action). A Proposed Action A_p is a tuple representing an action to be evaluated:

$$A_p = (id_A, name_A, params_A)$$

where id_A is a unique identifier, $name_A$ is a descriptive name, and $params_A$ is a set of parameters for the action.

Definition 1.31 (Verdict). A Verdict V is the outcome of the evaluation:

$$V = (m_V, J_V)$$

where m_V is the final deontic modality, and J_V is the justification, a set of id_τ of the Teloids that led to the verdict.

1.9.3 The Effect Ethos Structure (\mathbb{E})

The Effect Ethos engine, \mathbb{E} , comprises the norms and the relationships between them.

Definition 1.32 (Effect Ethos). The Effect Ethos \mathbb{E} is defined as a tuple:

$$\mathbb{E} = (S_\tau, I_{tag}, \mathcal{G}_\tau)$$

where:

- S_τ is the **Teloid Store**, a map from a Teloid ID to a Teloid: $S_\tau : id_\tau \mapsto \tau$.
- I_{tag} is the **Tag Index**, a map from a tag to a set of Teloid IDs: $I_{tag} : tag \mapsto \{id_{\tau_1}, id_{\tau_2}, \dots\}$.
- \mathcal{G}_τ is the **Teloid Graph**, a directed graph representing deontic relationships. It is a tuple (V_τ, E_τ) , where V_τ is the set of all id_τ in the store, and E_τ is a set of directed edges $(id_{\tau,i}, id_{\tau,j}, r)$. The relation $r \in \{Inherits, Defeats\}$ defines how norms interact.

1.9.4 The Deontic Inference Process ($\Pi_{Deontic}$)

The core dynamic of the Effect Ethos is the deontic inference process, $\Pi_{Deontic}$, which evaluates a proposed action.

Definition 1.33 ((Deontic Inference)). *The inference process is a mapping:*

$$\Pi_{Deontic} : (\mathbb{E}, A_p, \mathcal{C}_{sys}, Tags_{filter}) \rightarrow V$$

This process is executed in the following stages:

1. **Filtering:** A candidate set of Teloids, τ_{cand} , is selected from the Teloid Store S_τ by retrieving all Teloid IDs associated with the input $Tags_{filter}$ via the Tag Index I_{tag} .
2. **Activation:** A set of active Teloids, τ_{active} , is formed by evaluating the activation predicate p_τ for each $\tau \in \tau_{cand}$ against the Proposed Action A_p and the Context \mathcal{C}_{sys} .

$$\tau_{active} = \{\tau \in \tau_{cand} \mid p_\tau(\mathcal{C}_{sys}, A_p) = true\}$$

3. **Conflict Resolution:** A resolution function, $f_{resolve}$, is applied to the active set τ_{active} . This function traverses the Teloid Graph \mathcal{G}_τ and queries the Normative Ontologies to apply the DDIC heuristics (Lex Posterior, Lex Specialis, Lex Superior). Using the timestamp, specificity, and priority attributes of the Teloids, it resolves conflicts defined by *Defeats* edges to yield a final, conflict-free set of justified norms, τ_{final} .
4. **Verdict Finding:** A consensus function, $f_{consensus}$, determines the final verdict V from τ_{final} . It applies a strict precedence order to the modalities: Impermissible > Obligatory > Optional. The justification J_V consists of the IDs of all Teloids in τ_{final} .

1.10 EPP Example: Smoking, Tar, and Cancer

To illustrate the core concepts of the Effect Propagation Process (EPP) formalism in a more concrete manner, we consider the well-known causal chain: Smoking \rightarrow Tar in Lungs \rightarrow Lung Cancer. This example demonstrates how Causaloids, Contexts (implicitly for this simplification), and a CausaloidGraph can represent this system, and how effect propagation leads to an inference. This example is simplified and inspired by the structure typically modeled as a Directed Acyclic Graph (DAG) in classical causal inference [3], here adapted to EPP principles.

1.10.1 Defining the Causal Problem

We aim to model a system to infer the likelihood of lung cancer based on smoking habits, mediated by tar accumulation. The core causal hypotheses are:

1. Smoking (represented by nicotine levels) leads to increased tar in the lungs.
2. Increased tar in the lungs leads to a higher risk of lung cancer.

We will represent these as individual Causaloids within a CausaloidGraph. For simplicity, any contextual dependencies (e.g., thresholds for "high" nicotine or tar) are assumed to be encapsulated within the Causaloid functions themselves or drawn from an implicitly defined context for this illustrative purpose. Observations will be represented as numerical values.

1.10.2 Formalizing the Components

The Causaloid (chi)

We define two primary Causaloids:

- $\chi_{S \rightarrow T}$ (**Smoking \rightarrow Tar**):
 - $id_{\chi_{S \rightarrow T}} = 1$ (a unique identifier)
 - $type_{\chi_{S \rightarrow T}} = \text{Singleton}$
 - $f_{\chi_{S \rightarrow T}} : \mathcal{O}_{nicotine} \rightarrow \{\text{true}, \text{false}\}$, where $\mathcal{O}_{nicotine}$ is a numerical input representing nicotine level. The function $f_{\chi_{S \rightarrow T}}$ evaluates to true if the nicotine level exceeds a predefined threshold (e.g., 0.55), indicating a significant likelihood of tar presence due to smoking.
 - $\mathcal{C}_{refs} = \emptyset$ (assuming context-free logic for this example, or that thresholds are part of $f_{\chi_{S \rightarrow T}}$).
 - $desc_{\chi_{S \rightarrow T}} = \text{"Causal relation between smoking and tar in the lung."}$

- $\mathcal{A}_{linked}, I_{linked} = \text{Optional}$, representing linkage to foundational assumptions or prior inferences regarding this relationship.
- $\chi_{T \rightarrow C}$ (**Tar** \rightarrow **Cancer**):
 - $id_{\chi_{T \rightarrow C}} = 2$
 - $\text{type}_{\chi_{T \rightarrow C}} = \text{Singleton}$
 - $f_{\chi_{T \rightarrow C}} : \mathcal{O}_{\text{tar}} \rightarrow \{\text{true}, \text{false}\}$, where \mathcal{O}_{tar} is a numerical input representing tar level. The function $f_{\chi_{T \rightarrow C}}$ evaluates to true if the tar level exceeds a predefined threshold (e.g., 0.55).
 - $\mathcal{C}_{refs} = \emptyset$.
 - $\text{desc}_{\chi_{T \rightarrow C}} = \text{"Causal relation between tar in the lung and lung cancer."}$
 - $\mathcal{A}_{linked}, I_{linked} = \text{Optional}$.

The shared underlying logic for f_{χ} in this example (input observation \geq threshold) is abstracted into each Causaloid's specific function.

The CausaloidGraph (G)

We construct a CausaloidGraph $G_{STC} = (V_G, E_G, ID_G, \text{Name}_G)$ to represent the causal chain.

- $V_G = \{v_{g1}, v_{g2}\}$, where:
 - $v_{g1} = (id_{g1}, \text{payload}_{g1} = \chi_{S \rightarrow T})$
 - $v_{g2} = (id_{g2}, \text{payload}_{g2} = \chi_{T \rightarrow C})$
- $E_G = \{e_{g1}\}$, representing the link from smoking/tar to tar/cancer. For this simple chain:
 - $e_{g1} = (V_{\text{source}} = \{v_{g1}\}, V_{\text{target}} = \{v_{g2}\}, \text{logic}_{e1})$
 - logic_{e1} : Specifies that the evaluation of v_{g2} is conditioned by, or follows, the evaluation of v_{g1} . In a simple direct propagation, if v_{g1} becomes active, this contributes to the conditions for evaluating v_{g2} . (In a computational system with distinct inputs for each step, this logic might simply define the sequence or dependency).
- ID_G, Name_G : Appropriate identifiers and names (e.g., "Smoking-Tar-Cancer Model").

This formal G_{STC} represents the structure. In a practical implementation, a collection of Causaloids (like a vector) reasoned over sequentially can instantiate such a linear graph.

1.10.3 Formalizing Effect Propagation (Π_{EPP})

Consider input observations $O_{trig} = \{(id_{\chi_{S \rightarrow T}}, o_{\text{nicotine}}), (id_{\chi_{T \rightarrow C}}, o_{\text{tar}})\}$, where o_{nicotine} is the observed nicotine level and o_{tar} is the observed tar level. Let the initial state be $S_G(v_{g1}) = \text{inactive}$, $S_G(v_{g2}) = \text{inactive}$.

The propagation $\Pi_{EPP}((G_{STC}, S_G, \emptyset, O_{trig})) \rightarrow S'_G$ conceptually proceeds as follows, reflecting a sequential evaluation for this chain:

1. **Evaluate $\chi_{S \rightarrow T}$ (node v_{g1}):** The function $f_{\chi_{S \rightarrow T}}$ is evaluated with its corresponding input o_{nicotine} from O_{trig} . If $f_{\chi_{S \rightarrow T}}(o_{\text{nicotine}}) = \text{true}$ (e.g., nicotine ≥ 0.55), then $S'_G(v_{g1}) = \text{active}$. Else, $S'_G(v_{g1}) = \text{inactive}$.
2. **Evaluate $\chi_{T \rightarrow C}$ (node v_{g2}):** The function $f_{\chi_{T \rightarrow C}}$ is evaluated with its corresponding input o_{tar} from O_{trig} . The evaluation of v_{g2} in this chained model might also be implicitly conditioned on v_{g1} being active if the overall inference requires the full chain to hold. If $f_{\chi_{T \rightarrow C}}(o_{\text{tar}}) = \text{true}$ (e.g., tar ≥ 0.55), then $S'_G(v_{g2}) = \text{active}$. Else, $S'_G(v_{g2}) = \text{inactive}$.
3. **Final Inference/System State:** The overall state of the system might be defined by a conjunction: Cancer risk is inferred if $S'_G(v_{g1}) = \text{active} \wedge S'_G(v_{g2}) = \text{active}$. This reflects whether the complete causal pathway from smoking, via tar, to cancer is deemed active for the given observations.

This illustrative propagation shows how individual Causaloid evaluations, based on specific inputs, contribute to the overall state of the causal model. The logic_e of the hyperedge e_{g1} here implies a dependency or sequential consideration in this chain.

1.10.4 Observations

This simplified example illustrates:

- **Causaloids** (χ) as operational units (Section 1.5.1), encapsulating specific causal functions (f_χ).
- A simple **CausaloidGraph** (G) structure (Section 1.5.3) representing the causal chain.
- The **Effect Propagation Process** (Π_{EPP}) (Section ??) as the evaluation of Causaloids in a sequence determined by the graph structure and input data.
- The **State of the CausaloidGraph** (S_G) (Definition ??) evolving based on these evaluations.

While this example omits explicit dynamic Contexts (C_{sys}) or a dynamic Generative Function (Φ_{gen}) for brevity, it demonstrates how the core EPP structural entities map to a well-understood causal scenario.

1.11 Discussion

The preceding sections have laid out a formal, set-theoretic definition of the Effect Propagation Process (EPP). This formalism is a direct response to the need for new conceptual and computational tools to understand causality in systems where classical assumptions of a fixed spatiotemporal background and linear temporal order are demonstrably insufficient. The motivation stems from both the frontiers of fundamental physics, where spacetime itself is considered emergent, and the practical challenges of modeling complex adaptive systems across various scientific and engineering domains.

The primary significance of the EPP formalism lies in its principled detachment from spacetime. By making Context (C) an explicit, definable, and potentially non-Euclidean, dynamic fabric, and by defining Causaloids (χ) as operational units of effect transfer independent of a priori temporal ordering, EPP achieves a level of generality that classical causal formalisms cannot. This enables the modeling of causality in non-physical or abstract domains, systems with complex multi-scale temporal dynamics and feedback loops, and, crucially, systems where causal relationships themselves can emerge, transform, or dissolve in response to evolving contexts (dynamic regime shifts). The recursive isomorphism of CausaloidGraphs (G) further enhances expressive power, allowing for modular construction of complex causal models, while the clear separation of causal logic, contextual data, and propagation mechanisms facilitates transparency.

This foundational formalism, while potent, has current limitations. The EPP does not, in itself, provide algorithms for automated causal discovery of its structures from raw data; it provides the language to represent and reason with such structures once hypothesized. Compared to established frameworks like Pearl's SCMs, EPP offers greater flexibility for cyclic and emergent systems but currently lacks an equivalent to the extensive *do*-calculus.

2 Conclusion

This paper has presented a first-principles mathematical formalization of the Effect Propagation Process (EPP), a philosophical framework designed to advance the concept of causality beyond its classical, spacetime-bound limitations. The formalism provides precise set-theoretic definitions for the core EPP entities—Contextoids, Context Hypergraphs, Causaloids, and CausaloidGraphs—and outlines the operational dynamics of effect propagation, the generative functions enabling structural evolution, and the mechanisms of the Causal State Machine.

The development of this formalism was motivated by the increasing inadequacy of conventional causal models when faced with the intricate realities of advanced system engineering: specifically, the challenges of non-Euclidean data representation, non-linear temporal structures with complex feedback, and the emergence of causal relationships themselves within dynamic systems. By fundamentally detaching causality from a presupposed spacetime and a fixed temporal order, and by introducing the Causaloid as a unified, operational unit of effect transfer, the EPP formalism offers a robust and more general language for describing causal phenomena. Its capacity to explicitly model context as a dynamic, potentially non-Euclidean fabric, and to accommodate the notion of a generative function from which both context and causal structures can emerge, represents a significant departure from existing approaches.

This work lays a rigorous foundation for the continued theoretical development of EPP and underpins its computational instantiation in systems like DeepCausality¹. While this paper focuses on the foundational definitions, future efforts will build upon this to explore specific algorithms, derive formal properties, and demonstrate empirical utility across diverse complex systems. Ultimately, the EPP formalism is offered as a coherent and extensible framework for advancing our ability to model, understand, and reason about the deeply interconnected and dynamically evolving causal fabric of the world.

¹<https://deepcausality.com>

References

- [1] Marvin Hansen. The effect propagation process: A philosophical framework for post-quantum causality. <https://github.com/deepcausality-rs/papers>, June 2025. Accessed: June 2025.
- [2] Taylor Olson, Roberto Salas-Damian, and Kenneth D. Forbus. A defeasible deontic calculus for resolving norm conflicts, 2024.
- [3] J Pearl. Causality: models, reasoning, and inference cambridge university press new york. *NY, USA*, 2000.

Glossary

This glossary provides definitions for the key formal terms and symbols used throughout this paper to describe the Effect Propagation Process (EPP) and its components.

Effect Propagation Process (EPP)

The overarching philosophical framework and the formalized dynamic process (Π_{EPP}) of effect transfer within a CausaloidGraph, conditioned by the Contextual Fabric.

Contextual Fabric (C)

The structured environment, composed of Context Hypergraphs, within which effects propagate and causal relationships are conditioned. It is formalized as a Context Collection C_{sys} .

Contextoid (v)

The atomic unit of contextual information, defined as a tuple $v = (id_v, payload_v, adj_v)$. It encapsulates data, time, space, or spacetime values. (See Definition 3.1)

Contextoid Payload ($payload_v$)

The tagged union representing the actual data, temporal, spatial, or spatiotemporal value held by a Contextoid.

Context Hypergraph (C)

A structured collection of Contextoids (V_C) and the N-ary relationships (Hyperedges E_C) between them, defined as $C = (V_C, E_C, ID_C, Name_C)$. (See Definition 3.2)

Context Collection (C_{sys})

A finite set of distinct Context Hypergraphs, $C_{sys} = \{C_1, C_2, \dots, C_k\}$, representing the total available contextual information. (See Definition 3.4)

Context Accessor ($ContextAccessor(C_{refs})$)

A functional interface providing read-access for Causaloids to a specified subset of Context Hypergraphs ($C_{refs} \subseteq C_{sys}$). (See Definition 3.5)

Causaloid (χ)

The fundamental, operational unit of causal interaction, defined as $\chi = (id_\chi, type_\chi, f_\chi, C_{refs}, desc_\chi, \mathcal{A}_{linked}, I_{linked})$. It encapsulates a testable mechanism for effect transfer. (See Definition 4.1)

Causaloid Type ($type_\chi$)

Specifies the structural nature of a Causaloid (Singleton, Collection, or Graph), determining how its causal function f_χ is realized.

Causal Function (f_χ)

The operational logic associated with a Causaloid χ , which maps input observations (\mathcal{O}_{type}) and accessed context to an activation status ($\{\text{true}, \text{false}\}$). (See Section 4.2.2)

CausaloidGraph (G)

A hypergraph, defined as $G = (V_G, E_G, ID_G, Name_G)$, whose nodes (V_G) typically encapsulate Causaloids and whose hyperedges (E_G) define pathways and logic for effect propagation. Supports recursive isomorphism. (See Definition 4.2)

Causal Node (v_g)

A node within a CausaloidGraph G , defined as $v_g = (id_g, payload_g)$, where $payload_g$ can be a Causaloid, a collection of Causaloids, or another CausaloidGraph.

Causal Hyperedge (e_g)

A hyperedge within a CausaloidGraph G , defined as $e_g = (V_{source}, V_{target}, logic_e)$, representing a directed functional relationship.

State of CausaloidGraph (S_G)

A function $S_G : V_G \rightarrow \{\text{active}, \text{inactive}\}$ mapping each causal node in a CausaloidGraph to its current activation status. (See Definition 4.3)

Effect (ε)

Primarily the activation state (active/inactive) of a Causaloid within EPP, and the transfer of this status or derived information. (See Section 5.1.1)

Input Observations/Triggers (O_{input}, O_{trig})

External data or events (O_{input} being the set of all possible, O_{trig} being the current set) that can initiate or influence the Effect Propagation Process. (See Definition 5.1)

EPP Transition Function (Π_{EPP})

The core dynamic function $\Pi_{EPP} : (G, S_G, \mathcal{C}_{sys}, O_{trig}) \rightarrow S'_G$, describing how the state of a Causaloid-Graph evolves. (See Definition 5.2)

Operational Generative Function (Φ_{gen})

A formal function or set of functions representing meta-level rules for the dynamic evolution of the Contextual Fabric (Φ_{gen_C}), CausaloidGraphs (Φ_{gen_G}), or both (Φ_{gen_Total}). (See Section 5.2)

Causal State Machine (CSM, M)

An operational component, defined by its set of state-action pairs \mathcal{SA}_M , that links recognized Causal States (q) to deterministic Causal Actions (a). (See Definition 6.3)

Causal State (q)

A specific condition within a CSM, $q = (id_q, data_q, \chi_q, version_q)$, whose activation is determined by its associated Causaloid χ_q . (See Definition 6.1)

Causal Action (a)

A deterministic operation within a CSM, $a = (exec_a, descr_a, version_a)$, triggered by an active Causal State. (See Definition 6.2)