A Theoretical Exploration of Hyperconcepts: Hyperfunctions, Hyperrandomness, Hyperdecision-Making, and Beyond (Including a Survey of Hyperstructures)

Takaaki Fujita [1] *

[1] Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan.

Abstract

This paper delves into the concepts of Hyperfunctions and n-Superhyperfunctions, extending classical set functions into higher-order frameworks. By generalizing properties such as randomness, monotonicity, recursion, and symmetry, it explores their mathematical foundations and potential applications across various disciplines.

Additionally, the paper investigates future directions, including Weak Hyperstructures, Weak Hypergraphs, Hypercontexts, Hypervariables, Powerset Convolutions, Hypermatrices, Hyperfields, Hyperlattices, Cognitive Hypermaps, Hyperdecision-making, and more. It evaluates the feasibility of extending these concepts into the domain of SuperHyperStructures.

Although primarily theoretical, this study lays a robust foundation for future research, showcasing the versatility of Hyperstructures in addressing complex, multi-layered challenges.

Keywords: Hyperstructure, Hyperfunction, Function, Power set
MSC 2010 classifications: 08A05: Structures, substructures (algebraic systems), 03E20: Other classical set theory

1 Introduction

1.1 Function Theory and Hyperfunction Theory

In mathematics, a function is a relation that assigns exactly one output value to each input from a specified domain, mapping inputs to outputs according to a well-defined rule [4, 372]. This paper focuses on the discussion of various types of functions. Many different kinds of functions have been studied in mathematics, including the following:

- *Linear Function:* A function of the form $f(x) = ax + b$, where a and b are constants [63, 267].
- *Logarithmic Function:* A function of the form $f(x) = \log_a(x)$, where $a > 0$ and $a \neq 1$ [290, 351].
- *Trigonometric Function:* Functions such as $\sin(x)$, $\cos(x)$, and $\tan(x)$, which are periodic and commonly used in geometry and physics [382, 473].
- *Submodular Function:* A set function $f : 2^\Omega \to \mathbb{R}$ satisfying $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq \Omega$ [90, 306, 521].
- *Symmetric Function:* A function $f(x_1, x_2, \ldots, x_n)$ that satisfies

  $$f(x_1, x_2, \ldots, x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$$

  for any permutation σ of the indices $1, 2, \ldots, n$ [5, 24].
- *Fuzzy Function:* A function $f : X \to [0, 1]$ that assigns a degree of membership in the range $[0, 1]$ to each element $x \in X$, commonly used in fuzzy set theory to address uncertainty and imprecision [145, 375, 577].
- *Set Function:* A function $f : 2^\Omega \to \mathbb{R}$ defined on the power set 2^Ω of a universal set Ω, often used to measure properties such as size, weight, or probability of subsets.

In addition, a function known as the *Hyperfunction* has been introduced [194, 254, 362]. A Hyperfunction maps elements to subsets, formally defined as:

$$f : S \to \mathcal{P}(S),$$

where $\mathcal{P}(S)$ is the powerset of S.

More recently, a generalization called the *n-Superhyperfunction* has been studied [497]. This concept extends the idea of a Hyperfunction by utilizing higher-order powersets, enabling deeper abstraction and flexibility in modeling hierarchical structures.

1.2 Hyperstructure and Superhyperstructure

This subsection introduces the concepts of Hyperstructure and Superhyperstructure, which are mathematical constructs developed to represent hierarchical structures. A *Hyperstructure* generalizes the classical notion of powersets, extending it into broader mathematical frameworks that serve as a foundation for modeling more complex systems [344, 498, 500, 501].

In various studies, Hyperstructures are commonly applied in group theory and algebra theory (cf. [10, 11, 19, 87, 123, 124, 138, 451, 541, 543]). However, this paper adopts a perspective aligned with the concept of Power Sets. While there are notable similarities in the foundational ideas, the focus here is on the set-theoretic viewpoint. Moreover, in algebra theory and related fields, concepts such as Weak Hyperstructures (Hv-structures) have also been studied [16–18, 114, 119, 139, 543].

Building on this foundation, a *Superhyperstructure* introduces the concept of n-th powersets, enabling iterative and hierarchical generalizations of Hyperstructures. These advanced constructs allow for deeper levels of abstraction and provide tools to address increasing complexity [498, 500, 501]. The relationship between powersets and superhyperstructures is illustrated in Figure 1.

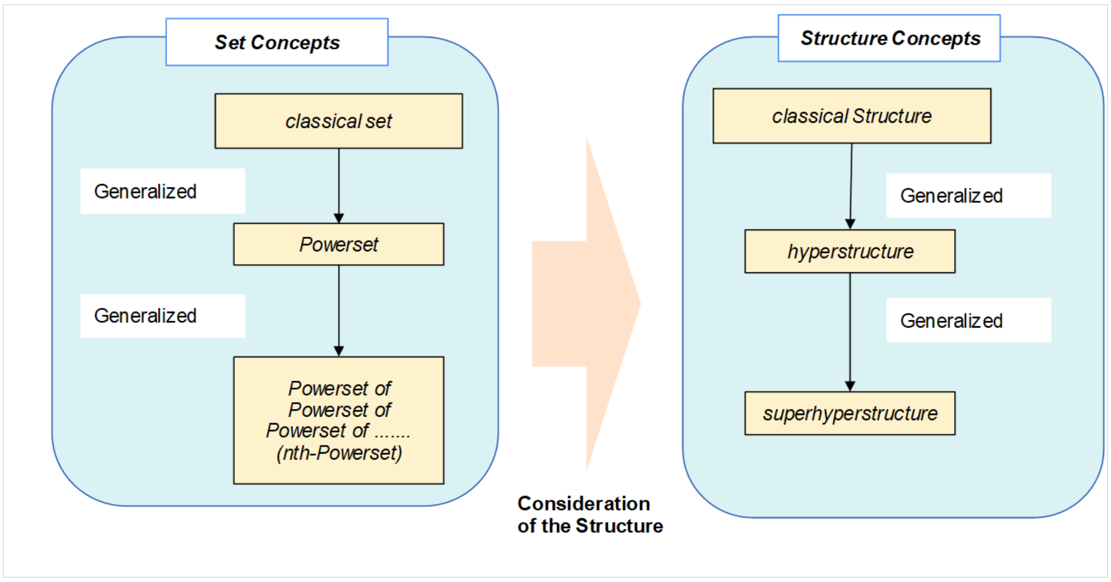

Figure 1: Relationships between Superhyperstructures and sets.

In graph theory [136], a *Hypergraph* is a generalization of traditional graphs where edges (called hyperedges) can connect more than two vertices [59,75,206,207], making it a prominent example of a Hyperstructure. Extending this concept, a *SuperHypergraph* incorporates advanced notions such as superedges and supervertices, offering a more abstract and flexible framework for representation (cf. [95, 172–174, 174, 178, 181–185, 198, 225, 226, 357, 441, 487, 488, 490, 493, 497, 497, 500]). Simply put, a SuperHypergraph can be viewed as a hierarchical and iterative extension of the Hypergraph concept.

Beyond graph theory, Hyperstructures and Superhyperstructures have found significant applications in various mathematical disciplines, including the following(cf. [176]):

- *Topology:* Topology studies properties of spaces preserved under continuous transformations, focusing on concepts like continuity, convergence, and connectedness [32,55,167,358]. The study of hypertopologies [133,331,333,370] and superhypertopologies [289,495,496,504] introduces innovative ways to explore topological properties.
- *Functional Analysis:* Hyperfunctions [254,362] and superhyperfunctions [492,497] expand the scope of function analysis to include multi-level interactions. This paper focuses on discussing these functions.
- *Soft Set Theory:* Soft Sets provide a parameterized framework for handling uncertainty [337, 359]. The development of hypersoft sets [1, 188, 235, 260, 367, 434, 458, 468, 491] and superhypersoft sets [179,283,494,505] extends classical soft set theory to handle more complex data structures.
- *Algebra:* Algebra studies mathematical symbols, operations, and rules for manipulating and solving equations [40, 288, 300]. Advances in hyperalgebras [120, 129, 251, 385, 437, 516, 538] and superhyperalgebras [261, 262, 312, 480, 489, 504] provide new perspectives on algebraic systems.
- *Automata Theory:* Automata are mathematical models of computation, describing abstract machines that process inputs via states [244, 247, 454]. Hyperautomata [70] and superhyperautomata [189] extend the traditional automata framework by integrating hyperstructural concepts.
- *Language:* In automata theory, a language is a set of strings formed from a specified alphabet [6,126,435]. Related concepts include Natural Language [431,548,564] and Large Language Models [102,549,574]. HyperLanguage [70, 71, 165, 180, 440] and SuperHyperLanguage [176] are well-known extensions.
- *Fuzzy Set Theory:* A fuzzy set assigns each element a membership degree between 0 and 1, representing partial inclusion [568–573]. Hyperfuzzy sets [199, 274, 508] and superhyperfuzzy sets [178] generalize fuzzy set theory to higher-order structures.
- *Ring Theory:* Hyperring theory [9, 30, 272] and superhyperring theory [499] enrich the study of commutative and non-commutative algebraic systems.
- *Rough Set Theory:* Rough Sets model uncertainty by approximating a target set using lower and upper bounds, focusing on indiscernibility within an equivalence relation [405–411]. Hyperrough sets [178, 478] and superhyperrough sets [178] extend rough set theory through hyperstructural frameworks.
- *Group Theory:* A group is a set with a binary operation satisfying closure, associativity, identity, and invertibility [86, 345, 354, 386, 464]. The evolution from hypergroups [281, 537, 540, 542] to superhypergroups [281] broadens the landscape of group theory.
- *Neutrosophic Sets:* Neutrosophic Sets generalize classic and fuzzy sets by incorporating truth, indeterminacy, and falsity membership degrees, offering flexibility for uncertain or inconsistent data [481–483, 502]. Hyperneutrosophic sets [127, 178] and superhyperneutrosophic sets [178] address uncertainties in higher-order systems.
- *Plithogenic Sets:* Plithogenic Sets extend neutrosophic sets by considering multiple attributes with contradictory, dependent, or independent criteria, enabling complex decision-making and advanced uncertainty modeling [187, 485, 486, 503]. HyperPlithogenic Sets [178] and SuperHyperPlithogenic Sets [178] address uncertainties in higher-order systems.

In addition to the concepts mentioned above, many other notions are well-known, including Hypernetwork (cf. [13,99,216,422]), Molecular Hypergraph (cf. [275,276]), Hypertree (cf. [197,206,224]), Hypervector (cf. [128,129,363]), HyperGroupoid (cf. [236,364,383,518]), SemiHyperGroup (cf. [72,291,346]), Hyperlattice(cf. [29, 233, 234]) and Hyperweighted Set (cf. [178]).

For reference, the relationships between Superhyperstructures and specific concepts are illustrated in Figure 2.

These examples illustrate the versatility of Hyperstructures and Superhyperstructures in enhancing our understanding of both mathematical theory and interdisciplinary applications. Their iterative and hierarchical frameworks make them invaluable tools for advancing research in areas requiring higher-level abstractions.

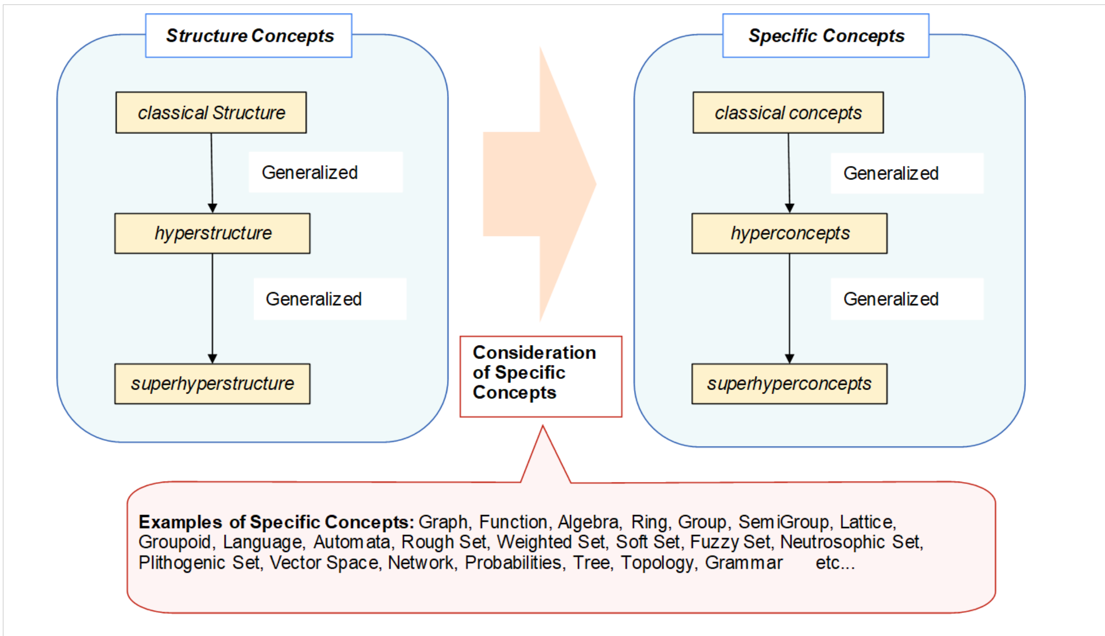

Figure 2: Relationships between Superhyperstructures and specific concepts.

1.3 Our Contribution in This Paper: Hyperfunctions and Various Hyperconcepts

This subsection provides a comprehensive overview of the contributions made in this paper.

In addition to the theoretical significance and the diversity of functions discussed earlier, it is important to highlight that these functions have been extensively studied for their applications across various fields. Within this context, research on Hyperfunctions and n-Superhyperfunctions emerges as increasingly relevant. However, existing studies on Hyperfunctions and n-Superhyperfunctions are currently limited.

To address this gap, this paper explores the mathematical extension of various functions within the frameworks of Hyperfunctions and n-Superhyperfunctions. These extensions provide a foundation for theoretically generalizing other types of functions, offering a versatile approach to modeling complex hierarchical relationships. By presenting this work, we aim to encourage broader adoption and further development of Hyperfunction and n-Superhyperfunction frameworks.

Specifically, the following functions are introduced and discussed:

- *Submodular Hyperfunction and n-Superhyperfunction:* Define hierarchical dependencies under submodular constraints at multiple levels.
- *Symmetric Hyperfunction and n-Superhyperfunction:* Ensure decisions remain invariant under permutations of elements.
- *Monotone Hyperfunction and n-Superhyperfunction:* Maintain non-decreasing properties across hierarchical decision frameworks.
- *Modular Hyperfunction and n-Superhyperfunction:* Represent additive relationships within nested decision structures.
- *Recursion Hyperfunction and n-Superhyperfunction:* Apply recursive logic for resolving hierarchical decision dependencies.
- *Random Hyperfunction and n-Superhyperfunction:* Incorporate probabilistic uncertainty into multi-layered decision processes.

Additionally, this paper explores future research directions, including Weak Hyperstructures, Weak Hypergraphs, Hypercontexts, Hypervariables, Powerset Convolutions, Hypermatrices, and Cognitive Hypermaps. Specifically, the following concepts are introduced in this study.

- *Weak Hyperstructures and n-Weak Superhyperstructures:* Generalize Hyperstructures with less strict axioms and constraints.
- *Hypercontext and Superhypercontext:* Represent layered environments influencing decisions and relationships across hierarchies.
- *HyperVariable and SuperHyperVariable:* Extend variables with multi-dimensional attributes across n-Superhyperstructures.
- *n-th Powerset Convolution:* Model hierarchical relationships combining constraints and criteria in higher-dimensional spaces.
- *HyperMatrix and SuperHyperMatrix:* Introduce matrices with hyperedges and recursive structures for complex computations.
- *Cognitive HyperMap and Cognitive SuperHyperMap:* Integrate multi-level cognitive processes into hypermap-based decision models.
- *Hyperfield and SuperHyperfield:* Expand algebraic fields to include multi-dimensional and hierarchical operational rules.
- *Hypermodules and Superhypermodules:* Generalize modules with hyperoperations over hierarchical structures$n$-Superhyperstructure.
- *Hyperlattices and Superhyperlattices:* Represent layered lattice frameworks with hyperoperations and extended properties.
- *Boolean Hyperalgebra and Superhyperalgebra:* Extend Boolean algebra with hyperoperations for hierarchical logical systems.
- *Project Management and Program Management related to Superhyperstructure:* Model multi-layered decision-making in management hierarchies using superhyperstructures.
- *HyperGame Theory and Superhypergame Theory:* These are generalized concepts of game theory.
- *Hyperdecision-making and Superhyperdecision-making:* Capture decision-making processes spanning multi-dimensional hierarchical frameworks.

It is crucial to emphasize that this paper primarily centers on theoretical generalizations. The practical feasibility and robustness of these methods for real-world applications necessitate further computational experimentation and validation.

1.4 The Structure of the Paper

The format of this paper is described below.

2 Preliminaries and Definitions

This section outlines the essential preliminaries and definitions required for the paper. While we aim to cover the core concepts, it is beyond the scope of this work to exhaustively define every term. Readers seeking further clarification are encouraged to consult the relevant literature for additional details.

2.1 Basic Set Theory

This subsection provides an overview of fundamental principles in set theory. For an in-depth exploration, we recommend referring to established references [242, 264, 270].

Definition 2.1 (Set)**.** [264] A *set* is a precisely defined collection of unique objects, known as *elements*. For any object x, it is always possible to determine definitively whether x is an element of the set. If A represents a set and x is an element within A, this is denoted as $x \in A$. Sets are typically written with curly braces. For instance, $A = \{1, 2, 3\}$ denotes a set containing the elements 1, 2, and 3.

Definition 2.2 (Subset)**.** [264] Given two sets A and B, A is defined as a *subset* of B, written $A \subseteq B$, if every element of A is also an element of B. Formally:

$$A \subseteq B \iff \forall x \, (x \in A \implies x \in B).$$

When $A \subseteq B$ but $A \neq B$, A is referred to as a *proper subset* of B, denoted by $A \subset B$.

Definition 2.3 (Empty Set)**.** [264] The *empty set*, denoted by $\emptyset$, is the unique set that contains no elements. Formally:

$$\forall x\, (x \notin \emptyset).$$

For example, $\emptyset = \{\}$.

Definition 2.4 (Universe Set)**.** [264] The *universe set*, denoted by U, is the set that contains all objects under consideration within a given context. Every set being studied is a subset of U. Formally:

$$A \subseteq U \quad \text{for all sets } A.$$

Definition 2.5 (Finite Set)**.** [264] A *finite set* is a set with a countable number of elements. Formally, it is defined as:

$$S = \{a_1, a_2, \ldots, a_n\},$$

where n is a finite integer.

Definition 2.6 (Operation)**.** (cf. [264]) An *operation* is a rule or function that combines elements of a set S to produce another element of S. Formally, an operation $\circ$ on a set S is a mapping:

$$\circ : S \times S \to S.$$

For example, addition and multiplication are operations on the set of real numbers $\mathbb{R}$.

Definition 2.7 (Binary Operation)**.** (cf. [85]) A *binary operation* on a set S is a function $* : S \times S \to S$ that combines any two elements $a, b \in S$ to produce another element $a * b \in S$.

Definition 2.8 (Set Union)**.** [264] The *union* of two sets A and B, denoted $A \cup B$, is the set containing all elements that are in A, B, or both. Formally:

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}.$$

Definition 2.9 (Set Intersection)**.** [264] The *intersection* of two sets A and B, denoted $A \cap B$, is the set containing all elements common to A and B. Formally:

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}.$$

2.2 Hyperstructure and Superhyperstructure

This subsection introduces the concepts of Hyperstructure and Superhyperstructure. A *Hyperstructure* is a mathematical framework built upon the structure of a powerset, while a *Superhyperstructure* generalizes this concept by incorporating the n-th powerset. This extension facilitates the representation of multi-layered hierarchical systems [171,500,501], providing a robust foundation for modeling increasingly complex relationships. For a clear understanding of the basic ideas, readers are encouraged to refer to [500] as needed. The formal definition of the n-th powerset is presented below.

Definition 2.10 (Base Set)**.** A *base set* is a primary set S from which more elaborate constructs, such as powersets and hyperstructures, are generated. Formally, it is defined as:

$$S = \{x \mid x \text{ is a member of the specified domain}\}.$$

All elements of derived structures like $\mathcal{P}(S)$ or $\mathcal{P}_n(S)$ are ultimately drawn from the elements of S.

Definition 2.11 (Powerset)**.** [175,446] The *powerset* of a set S, written $\mathcal{P}(S)$, is the collection of all subsets of S, including the empty set and S itself. Formally, it is expressed as:

$$\mathcal{P}(S) = \{A \mid A \subseteq S\}.$$

Proposition 2.12. *The powerset generalizes the concept of a set.*

Proof. This is self-evident from the definition. □

Definition 2.13 (n-th Powerset)**.** (cf. [175, 480, 500]) The n-th powerset of a set H, denoted as $P_n(H)$, is defined iteratively. Starting with the standard powerset, the n-th powerset is constructed as follows:

$$P_1(H) = P(H), \quad P_{n+1}(H) = P(P_n(H)), \quad \text{for } n \geq 1.$$

Similarly, the n-th non-empty powerset of H, represented by $P_n^*(H)$, is defined recursively as:

$$P_1^*(H) = P^*(H), \quad P_{n+1}^*(H) = P^*(P_n^*(H)).$$

Here, $P^*(H)$ represents the powerset of H excluding the empty set.

Proposition 2.14. *(cf. [175, 480, 500]) The n-th powerset extends the concept of a standard powerset.*

Proof. This is immediately clear from the definition of the n-th powerset, as it involves repeated applications of the powerset operation. □

To formally define Hyperstructures and Superhyperstructures, we proceed as follows.

Definition 2.15 (Classical Structure)**.** (cf. [480,500]) A *Classical Structure* is a mathematical framework built on a non-empty set H, equipped with one or more *Classical Operations* and satisfying certain *Classical Axioms*. The structure is defined as follows:

A *Classical Operation* is a mapping:

$$\#_0 : H^m \to H,$$

where $m \geq 1$ is an integer, and H^m represents the m-fold Cartesian product of H. These operations may include, for example, addition or multiplication in algebraic structures such as groups, rings, or fields.

Definition 2.16 (Hyperstructure)**.** (cf. [175,480,500]) A *Hyperstructure* is a mathematical framework built on the powerset of a base set. It is formally described as:

$$\mathcal{H} = (\mathcal{P}(S), \circ),$$

where S is the base set, $\mathcal{P}(S)$ denotes its powerset, and $\circ$ is an operation defined on elements of $\mathcal{P}(S)$.

Proposition 2.17. *Every hyperstructure serves as a generalization of a classical structure.*

Proof. This is evident. □

Proposition 2.18. *A Hyperstructure is inherently characterized by the structure of a powerset.*

Proof. This property directly arises from the definition of a Hyperstructure, which is built upon the powerset $\mathcal{P}(S)$. □

Definition 2.19 (n-Superhyperstructure)**.** (cf. [480, 500]) An *n-Superhyperstructure* is a generalization of a Hyperstructure achieved through n-fold iterations of the powerset operation. Formally, it is represented as:

$$\mathcal{SH}_n = (\mathcal{P}_n(S), \circ),$$

where S is the base set, $\mathcal{P}_n(S)$ is the n-th powerset of S, and $\circ$ is a general operation defined on $\mathcal{P}_n(S)$.

Proposition 2.20. *An n-Superhyperstructure is characterized by the structure of the n-th powerset.*

Proof. This result is a direct consequence of the definition of an n-Superhyperstructure, which is explicitly built using the n-th powerset. □

Proposition 2.21. *Every n-Superhyperstructure serves as a generalization of a Hyperstructure.*

Proof. By definition, a Hyperstructure is based on the powerset $\mathcal{P}(S)$, which corresponds to the 1-th powerset $\mathcal{P}_1(S)$. Since an n-Superhyperstructure uses the n-th powerset $\mathcal{P}_n(S)$, it inherently generalizes the Hyperstructure when $n > 1$. □

2.3 Hyperfunction and n-Superhyperfunction

In the context of Hyperstructure and n-Superhyperstructure in functions, the concepts of Hyperfunction and n-Superhyperfunction are well-known [254,362]. Hyperfunctions, in particular, have been extensively studied, including various applications. Relevant definitions and theorems are presented below.

Definition 2.22 (Function)**.** (cf. [68,223]) A *function* is a mathematical relation that maps each element x in a set X (the domain) to exactly one element y in another set Y (the codomain), denoted as $f : X \to Y$.

Definition 2.23 (Hyperoperation)**.** (cf. [443, 534–536]) A *hyperoperation* is a generalization of a binary operation where the result of combining two elements is a set, not a single element. Formally, for a set S, a hyperoperation $\circ$ is defined as:

$$\circ : S \times S \to \mathcal{P}(S),$$

where $\mathcal{P}(S)$ is the powerset of S.

Definition 2.24 (Hyperfunction)**.** [492, 497] A *Hyperfunction* is a function where the domain remains a classical set S, but the codomain is extended to the powerset of S, denoted $\mathcal{P}(S)$. Formally, a Hyperfunction f is defined as:

$$f : S \to \mathcal{P}(S).$$

For any $x \in S$, $f(x) \subseteq S$ is a subset of S. This allows the function to map an element of S to multiple elements, enabling greater flexibility compared to classical functions.

Example 2.25. Let $S = \{1, 2\}$. Define $f : S \to \mathcal{P}(S)$ as follows:

$$f(1) = \{1, 2\}, \quad f(2) = \{2\}.$$

Here, $f(1)$ maps to a subset of S, $\{1, 2\}$, and $f(2)$ maps to $\{2\}$. Both outputs belong to $\mathcal{P}(S)$.

Proposition 2.26. *A Hyperfunction $f : S \to \mathcal{P}(S)$ forms a Hyperstructure.*

Proof. By definition, the codomain of a Hyperfunction f is the powerset $\mathcal{P}(S)$. A Hyperstructure is defined as a mathematical construct based on the powerset $\mathcal{P}(S)$. Therefore, the Hyperfunction inherently operates within the framework of a Hyperstructure. □

Proposition 2.27. *A Hyperfunction generalizes a Function.*

Proof. This follows directly from the definition. □

Definition 2.28 (SuperHyperOperations)**.** [500] Let H be a non-empty set, and let $P(H)$ be the powerset of H. Define the n-th powerset $P^n(H)$ recursively:

$$P^0(H) = H, \quad P^{k+1}(H) = P(P^k(H)) \text{ for } k \geq 0.$$

A *SuperHyperOperation* of order (m, n) is an m-ary operation:

$$\circ^{(m,n)} : H^m \to P^n_*(H)$$

where $P^n_*(H)$ denotes the n-th powerset of H possibly excluding the empty set (for a classical-type SuperHyperOperation) or including it (for a Neutrosophic-type SuperHyperOperation).

If the codomain is $P^n_*(H)$ without the empty set, we call it a *classical-type (m,n)-SuperHyperOperation*. If the codomain is $P^n(H)$ including the empty set, we call it a *Neutrosophic (m,n)-SuperHyperOperation*.

In either case, these SuperHyperOperations are higher-order generalizations of hyperoperations, capturing multi-level complexity through n-th powerset constructions.

Definition 2.29 (n-Superhyperfunction)**.** An *n-Superhyperfunction* generalizes the concept of a Hyperfunction by using the n-th powerset $\mathcal{P}_n(S)$ as the codomain. Formally, for $n \geq 2$, an n-Superhyperfunction f is defined as:

$$f : \mathcal{P}_r(S) \to \mathcal{P}_n(S),$$

where $0 \leq r \leq n$, and $\mathcal{P}_n(S)$ is the n-th powerset of S. This definition allows f to map subsets of S (from $\mathcal{P}_r(S)$) to elements in the n-th powerset $\mathcal{P}_n(S)$.

Example 2.30. Let $S = \{1, 2\}$. The 2nd powerset $\mathcal{P}_2(S)$ of S is:

$$\mathcal{P}_2(S) = \{\emptyset, \{\emptyset\}, \{\{1\}\}, \{\{2\}\}, \{\{1, 2\}\}, \{\emptyset, \{1\}\}, \dots, \mathcal{P}(S)\}.$$

Define $f : \mathcal{P}(S) \to \mathcal{P}_2(S)$ as:

$$f(\{1\}) = \{\{1\}, \{2\}\}, \quad f(\{2\}) = \{\{\emptyset\}, \{1, 2\}\}.$$

Here, f maps subsets of S (from $\mathcal{P}(S)$) to elements of $\mathcal{P}_2(S)$, fulfilling the definition of a 2-Superhyperfunction.

Proposition 2.31. *An n-Superhyperfunction $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ is associated with an n-Superhyperstructure.*

Proof. By definition, an n-Superhyperstructure is constructed using the n-th powerset $\mathcal{P}_n(S)$. Since the codomain of an n-Superhyperfunction is $\mathcal{P}_n(S)$, the function operates within the framework of an n-Superhyperstructure. Hence, f is inherently tied to the n-Superhyperstructure. □

Theorem 2.32. *Every n-Superhyperfunction generalizes both classical functions and Hyperfunctions:*

- *For $r = n = 0$, $f : S \to S$, recovering the definition of a classical function.*
- *For $r = 0$ and $n = 1$, $f : S \to \mathcal{P}(S)$, corresponding to a Hyperfunction.*
- *For $r = 0$ and $n \geq 2$, $f : S \to \mathcal{P}_n(S)$, representing an n-Superfunction.*

Proof. The structure of $\mathcal{P}_n(S)$ recursively incorporates the elements of $\mathcal{P}_{n-1}(S)$. When $n = 0$, $\mathcal{P}_0(S) = S$, recovering classical functions. For $n = 1$, $\mathcal{P}_1(S) = \mathcal{P}(S)$, yielding Hyperfunctions. For $n \geq 2$, $\mathcal{P}_n(S)$ iterates the powerset operation, resulting in n-Superfunctions. This hierarchy ensures that n-Superhyperfunctions encompass all lower-order functions. □

The relationship between functions and superhyperstructures is illustrated in Figure 3.

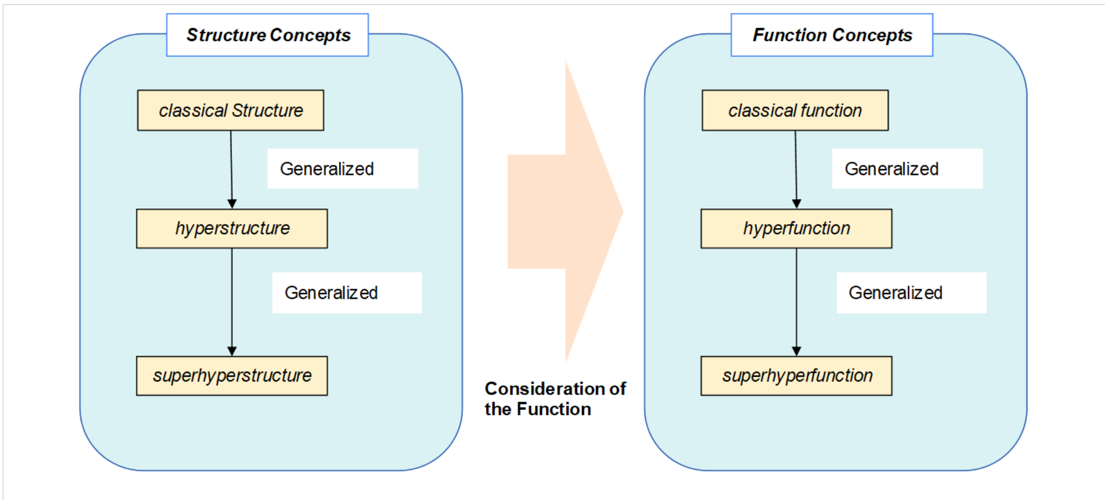

Figure 3: Relationships between Superhyperstructures and functions.

3 Results Presented in This Paper: Hyperfunctions

This section highlights the main contributions of this paper, where we extend several well-known functions to the frameworks of Hyperfunctions and n-Superhyperfunctions. Furthermore, we explore their relationships with other related concepts. It is worth noting that this paper primarily focuses on theoretical generalizations. The practical feasibility and robustness of these methods in real-world applications remain to be investigated through further computational experiments and validation efforts.

3.1 Submodular Hyperfunction and Their Generalizations

A submodular function exhibits a diminishing returns property, where adding an element has less impact as the set grows [48, 90, 100, 107, 306, 387, 521]. Submodular functions are applied in optimization [98, 528], machine learning [66, 66, 258], sensor placement [526, 527], and network design [132, 470, 471].

Definition 3.1 (Real Number)**.** (cf. [147, 277, 442]) A *real number* is any value that can represent a distance along a continuous line, including all rational and irrational numbers. The set of real numbers is denoted by $\mathbb{R}$.

Definition 3.2 (Submodular Function with Bounds)**.** [90, 521] A function $f : 2^{\Omega} \to \mathbb{R}$ is called *submodular* if for all $A, B \subseteq \Omega$, the following inequality holds:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B).$$

In addition, the function may satisfy boundary conditions such as:

$$f(\emptyset) = 0, \quad \text{and} \quad f(\Omega) \leq C,$$

where $C \geq 0$ is a constant. These constraints can provide normalization or ensure the function meets specific feasibility criteria.

Example 3.3. Let $\Omega = \{1, 2, 3\}$, and define $f : 2^{\Omega} \to \mathbb{R}$ as:

$$f(A) = |A|^2.$$

To verify submodularity:

- Compute $f(\{1\}) + f(\{2\}) = 1 + 1 = 2$ and $f(\{1, 2\}) + f(\emptyset) = 4 + 0 = 4$. It is evident that submodularity fails since $f(A) = |A|^2$ does not satisfy the diminishing returns property.

Now, redefine f as:

$$f(A) = \min(|A|, 1).$$

For this revised f:

- If either A or B is empty, submodularity holds trivially because $f(\emptyset) = 0$.
- For any $A, B \subseteq \Omega$, $f(A) + f(B)$ is always greater than or equal to $f(A \cup B) + f(A \cap B)$, since $f(A)$ takes values either 0 (if $A = \emptyset$) or 1 (if $A \neq \emptyset$).

Thus, $f(A) = \min(|A|, 1)$ is a valid example of a submodular function.

Theorem 3.4. *A Submodular Function is a Function.*

Proof. This follows directly from the definition. □

Definition 3.5 (Submodular Hyperfunction)**.** A *submodular hyperfunction* is a function:

$$f : 2^{2^{\Omega}} \to \mathbb{R}$$

such that for all $A, B \subseteq 2^{\Omega}$:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B).$$

An analogous bounding condition could be:

$$f(\emptyset) = 0, \quad f(\{\Omega\}) \leq D,$$

for some constant $D \geq 0$.

Example 3.6 (Submodular Hyperfunction)**.** Let $\Omega = \{1, 2\}$, so $2^{\Omega} = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Define $f : 2^{2^{\Omega}} \to \mathbb{R}$ by:

$$f(X) = \begin{cases} 0 & \text{if } X = \emptyset, \\ |X| & \text{otherwise.} \end{cases}$$

For any $A, B \subseteq 2^{\Omega}$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ holds since f is essentially counting subsets. Checking a few cases confirms submodularity. For example: Let $A = \{\{1\}\}$, $B = \{\{2\}\}$. Then $f(A) = f(B) = 1$, $f(A \cup B) = f(\{\{1\}, \{2\}\}) = 2$, $f(A \cap B) = f(\emptyset) = 0$. Thus $f(A) + f(B) = 2$, $f(A \cup B) + f(A \cap B) = 2 + 0 = 2$, inequality holds.

Theorem 3.7. *A Submodular Hyperfunction generalizes the concept of a Submodular Function.*

Proof. Let $f : 2^{2^{\Omega}} \to \mathbb{R}$ be a Submodular Hyperfunction. For a classical Submodular Function $g : 2^{\Omega} \to \mathbb{R}$, consider the special case where f is restricted to subsets of 2^{Ω} consisting of singleton sets, i.e., $A = \{\{a\}\}$ for $a \in \Omega$. Define $g(A) = f(\{\{a\}\})$.

Now verify the submodularity condition:

$$g(A) + g(B) \geq g(A \cup B) + g(A \cap B).$$

Using the definition of f, this becomes:

$$f(\{\{a\}\}) + f(\{\{b\}\}) \geq f(\{\{a\}, \{b\}\}) + f(\{\{a\} \cap \{b\}\}),$$

which simplifies to:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B),$$

because $A, B \subseteq 2^{\Omega}$. Thus, f restricted to singletons satisfies the submodularity condition, demonstrating that Submodular Hyperfunctions generalize Submodular Functions. □

Theorem 3.8. *A Submodular Hyperfunction inherits the structural properties of a Hyperfunction.*

Proof. By definition, a Hyperfunction is a mapping:

$$f : S \to \mathcal{P}(S),$$

where S is the domain and $\mathcal{P}(S)$ is the powerset of S. Extending this, a Submodular Hyperfunction is defined as:

$$f : 2^{2^{\Omega}} \to \mathbb{R}.$$

Here, $2^{2^{\Omega}}$ is the powerset of 2^{Ω}, aligning with the structure of a Hyperfunction. For any subsets $A, B \subseteq 2^{\Omega}$, $f(A)$ and $f(B)$ represent mappings into $\mathbb{R}$, and the operations $A \cup B$ and $A \cap B$ occur within $2^{2^{\Omega}}$, preserving the structural framework of a Hyperfunction.

Thus, the Submodular Hyperfunction retains the essential features of a Hyperfunction while imposing additional submodularity constraints on f. □

We now generalize the domain further to $P^n(\Omega)$.

Definition 3.9 (Submodular n-Superhyperfunction)**.** For $n \geq 1$, a function:

$$f : P^n(\Omega) \to \mathbb{R}$$

is *submodular* if for all $X, Y \subseteq P^n(\Omega)$:

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y).$$

Bounding conditions might be:

$$f(\emptyset) = 0, \quad f(P^n(\Omega)) \leq E,$$

for some $E \geq 0$.

Example 3.10. Consider $\Omega = \{1\}$. The recursive construction of powersets gives:

- $P^1(\Omega) = 2^\Omega = \{\emptyset, \{1\}\}$,
- $P^2(\Omega) = 2^{2^\Omega} = \{\emptyset, \{\emptyset\}, \{\{1\}\}, \{\emptyset, \{1\}\}\}$.

Define $f : P^2(\Omega) \to \mathbb{R}$ by:

$$f(X) = \begin{cases} 0 & X = \emptyset, \\ 1 & \text{otherwise.} \end{cases}$$

To verify submodularity, consider any $X, Y \subseteq P^2(\Omega)$:

- If $X = \emptyset$ or $Y = \emptyset$, then $f(X) + f(Y) = f(X \cup Y) + f(X \cap Y) = 1$, satisfying submodularity.
- If $X, Y \neq \emptyset$, then $f(X) + f(Y) = 1 + 1 = 2$, and $f(X \cup Y) + f(X \cap Y)$ equals 2, ensuring the inequality holds.

Thus, $f(X)$ is a Submodular n-SuperHyperfunction.

Theorem 3.11. *A Submodular n-SuperHyperfunction possesses the structural properties of an n-SuperHyperfunction.*

Proof. By definition, an n-SuperHyperfunction is a mapping:

$$f : \mathcal{P}_r(S) \to \mathcal{P}_n(S),$$

where $\mathcal{P}_n(S)$ represents the n-th powerset of S. In the context of a Submodular n-SuperHyperfunction, the domain is extended to $P^n(\Omega)$, and the codomain is $\mathbb{R}$, with the additional submodularity condition:

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y), \quad \forall X, Y \subseteq P^n(\Omega).$$

The structure of $P^n(\Omega)$ is recursively defined, ensuring that $P^n(\Omega)$ is a valid input space for any n-SuperHyperfunction. Specifically:

- When $n = 1$, $P^1(\Omega) = 2^\Omega$, reducing the Submodular n-SuperHyperfunction to a Hyperfunction with submodular constraints.
- For $n > 1$, $P^n(\Omega)$ is the powerset of $P^{n-1}(\Omega)$, ensuring the input domain satisfies the structural requirements of an n-SuperHyperfunction.

The output space $\mathbb{R}$ of the Submodular n-SuperHyperfunction retains compatibility with the mapping properties of an n-SuperHyperfunction because the submodularity condition imposes a real-valued inequality rather than altering the function's structural behavior. Thus, a Submodular n-SuperHyperfunction adheres to the structural framework of an n-SuperHyperfunction while introducing additional constraints.

Therefore, a Submodular n-SuperHyperfunction possesses all the structural properties of an n-SuperHyperfunction. □

Theorem 3.12. *A submodular n-superhyperfunction generalizes both submodular functions and submodular hyperfunctions. More precisely:*

- *For $n = 1$, a submodular 1-superhyperfunction is exactly a submodular hyperfunction.*
- *For $n = 0$ (interpreting $f : 2^\Omega \to \mathbb{R}$ as a submodular function on the original ground set), we recover the classical notion of a submodular function.*

Proof. When $n = 0$, $P^0(\Omega) = \Omega$. A submodular 0-superhyperfunction would be defined on $P^0(\Omega) = \Omega$, but to align with standard definitions, we consider $f : 2^\Omega \to \mathbb{R}$ directly, yielding the classical submodular function definition.

When $n = 1$, $f : P^1(\Omega) = 2^\Omega \to \mathbb{R}$ defined with submodularity at the next level ($f : 2^{2^\Omega} \to \mathbb{R}$) matches the definition of a submodular hyperfunction.

Thus, the notion of an n-superhyperfunction naturally extends these concepts by increasing the structural complexity of the domain, and for lower n, it reduces to the previously known classes of submodular functions. □

3.2 Symmetric Hyperfunction and Their Generalizations

We now introduce the concept of symmetry. A function defined on 2^Ω is called symmetric if it remains invariant under any permutation of the ground set Ω. As a supplementary remark, structures that are not symmetric are sometimes referred to as *asymmetric* [151, 392].

Definition 3.13 (Symmetric Function)**.** (cf. [5, 24, 177]) A function $f : 2^\Omega \to \mathbb{R}$ is *symmetric* if for every bijection $\pi : \Omega \to \Omega$ and every $S \subseteq \Omega$, we have

$$f(S) = f(\pi(S)).$$

This implies that f depends only on the structure of subsets of Ω and not on the particular labeling of the elements.

Theorem 3.14. *A Symmetric Function is a Function.*

Proof. This follows directly from the definition. □

By analogy, we can extend symmetry to hyperfunctions and n-superhyperfunctions. For a hyperfunction $f : 2^{2^\Omega} \to \mathbb{R}$, a permutation π of the ground set Ω induces a permutation on 2^Ω defined by $\pi'(A) = \{\pi(x) : x \in A\}$ for $A \subseteq \Omega$. In turn, π' induces a permutation on 2^{2^Ω} by applying it to each element of a set of subsets.

Definition 3.15 (Symmetric Hyperfunction)**.** A hyperfunction $f : 2^{2^\Omega} \to \mathbb{R}$ is *symmetric* if for every bijection $\pi : \Omega \to \Omega$ and every $X \subseteq 2^\Omega$,

$$f(X) = f(\{\pi'(A) : A \in X\}),$$

where $\pi'(A) = \{\pi(a) : a \in A\}$.

Example 3.16. If $\Omega = \{1, 2\}$, let $f : 2^{2^\Omega} \to \mathbb{R}$ be defined by $f(X) = |X|$. Any permutation of $\{1, 2\}$ just relabels these elements, but since f depends only on the cardinality of X, f remains invariant. Thus it is symmetric.

Theorem 3.17. *A Symmetric Hyperfunction $f : 2^{2^\Omega} \to \mathbb{R}$ is associated with a Hyperstructure.*

Proof. A Hyperstructure is defined on the powerset $\mathcal{P}(\Omega)$, and a Symmetric Hyperfunction operates on the powerset 2^{2^Ω}, which is itself a powerset of $\mathcal{P}(\Omega)$. The symmetry condition ensures that f is invariant under relabeling (or permutations) of the elements of Ω. This invariance aligns with the structural properties of a Hyperstructure, which is also independent of specific labeling. Therefore, a Symmetric Hyperfunction inherently operates within the framework of a Hyperstructure. □

Similarly, for n-superhyperfunctions, a permutation π on Ω induces a permutation on $P^n(\Omega)$ level-by-level. We say:

Definition 3.18 (Symmetric n-Superhyperfunction)**.** A function $f : P^n(\Omega) \to \mathbb{R}$ is *symmetric* if for every bijection $\pi : \Omega \to \Omega$, the induced action of π on $P^n(\Omega)$ (applying π at the base level and lifting it through all n-levels of powersets) satisfies:

$$f(X) = f(\pi^{(n)}(X))$$

for all $X \in P^n(\Omega)$, where $\pi^{(n)}$ is the iterated induced permutation acting on elements of $P^n(\Omega)$.

Example 3.19. Let $f : P^2(\Omega) \to \mathbb{R}$ be defined as $f(X) = |X|$. For $\Omega = \{1, 2\}$, any permutation of $\{1, 2\}$ induces a re-labeling of subsets at the second level, but the sizes of these sets of subsets remain unchanged, thus f is symmetric.

Theorem 3.20. *A Symmetric n-Superhyperfunction $f : P^n(\Omega) \to \mathbb{R}$ is associated with an n-Superhyperstructure.*

Proof. An n-Superhyperstructure is defined on the n-th powerset $P^n(\Omega)$, and a Symmetric n-Superhyperfunction operates on $P^n(\Omega)$. The symmetry condition ensures that f is invariant under any permutation π of the elements of Ω, propagated across all n-levels of powerset iterations. This invariance aligns with the hierarchical and structural properties of an n-Superhyperstructure, which is independent of specific labeling at any level. Therefore, a Symmetric n-Superhyperfunction is inherently associated with an n-Superhyperstructure. □

Theorem 3.21. *Symmetric n-superhyperfunctions generalize symmetric functions and symmetric hyperfunctions. Specifically:*

- *For $n = 0$, symmetric 0-superhyperfunctions are just symmetric functions $f : 2^\Omega \to \mathbb{R}$.*
- *For $n = 1$, symmetric 1-superhyperfunctions correspond to symmetric hyperfunctions.*

Proof. The proof follows the same reasoning as the submodular case. Restricting to $n = 0$ recovers the standard definition of a symmetric function on 2^Ω. For $n = 1$, we recover symmetric hyperfunctions on 2^{2^Ω}. Thus, symmetry at level n generalizes the classical notion of symmetry at lower levels. □

Remark 3.22. Naturally, one can combine submodularity and symmetry. A submodular and symmetric n-superhyperfunction is a function $f : P^n(\Omega) \to \mathbb{R}$ that is both submodular as defined above and symmetric under any permutation of the base set Ω.

Remark 3.23. In the context of general submodular functions, a submodular set function that takes natural numbers instead of real numbers is sometimes referred to as a *connectivity function* [193, 393]. Additionally, there is a concept known as a *submodular partition function*, which incorporates the idea of partitions into submodular set functions [31].

It might be worth considering whether hyperfunction versions of these concepts could be defined or studied as needed. However, this paper omits such discussions.

3.3 Monotone Hyperfunction and Their Generalizations

A monotone function is a function that either consistently increases or decreases without reversing direction across its domain (cf. [21, 317, 438, 476, 563, 580]). These concepts can be extended to Hyperfunctions and n-Superhyperfunctions. Relevant definitions and theorems are provided below.

Definition 3.24 (Monotone Set Function)**.** (cf. [265,273,322]) Let Ω be a finite set. A set function $\mu : 2^\Omega \to \mathbb{R}$ is called *monotone* if for all $E, F \subseteq \Omega$ with $E \subseteq F$, we have:

$$\mu(E) \leq \mu(F).$$

Example 3.25. Consider $\Omega = \{1, 2, 3\}$ and define $\mu : 2^\Omega \to \mathbb{R}$ by $\mu(S) = |S|$. Since $E \subseteq F$ implies $|E| \leq |F|$, it follows that $\mu(E) \leq \mu(F)$, so μ is monotone.

Theorem 3.26. *A Monotone Set Function is a Function.*

Proof. This follows directly from the definition. □

We extend monotonicity to hyperfunctions, whose domain is 2^{2^Ω}.

Definition 3.27 (Monotone Set Hyperfunction)**.** A hyperfunction $\mu : 2^{2^\Omega} \to \mathbb{R}$ is *monotone* if for all $A, B \subseteq 2^\Omega$ with $A \subseteq B$, we have:

$$\mu(A) \leq \mu(B).$$

Example 3.28. Let $\Omega = \{1, 2\}$, so $2^{\Omega} = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Define $\mu : 2^{2^{\Omega}} \to \mathbb{R}$ by:

$$\mu(X) = |X|.$$

If $A \subseteq B$ for $A, B \subseteq 2^{\Omega}$, then $|A| \leq |B|$, hence $\mu(A) \leq \mu(B)$, making μ monotone.

Theorem 3.29. *A Monotone Set Hyperfunction $\mu : 2^{2^{\Omega}} \to \mathbb{R}$ is associated with a Hyperstructure.*

Proof. A Hyperstructure is defined on the powerset $\mathcal{P}(\Omega) = 2^{\Omega}$, and a Monotone Set Hyperfunction operates on $\mathcal{P}(\mathcal{P}(\Omega)) = 2^{2^{\Omega}}$, which is itself a powerset. The monotonicity property ensures that μ respects the ordering of subsets within $2^{2^{\Omega}}$. This property aligns with the foundational principles of a Hyperstructure, which relies on the organization and relationships of subsets within its powerset. Thus, a Monotone Set Hyperfunction naturally operates within a Hyperstructure. □

Theorem 3.30. *A Monotone Set Hyperfunction generalizes a Monotone Set Function.*

Proof. A Monotone Set Function $\mu : 2^{\Omega} \to \mathbb{R}$ is defined on the first-level powerset 2^{Ω}, whereas a Monotone Set Hyperfunction $\mu : 2^{2^{\Omega}} \to \mathbb{R}$ operates on the second-level powerset $2^{2^{\Omega}}$. By definition, any Monotone Set Function satisfies $\mu(E) \leq \mu(F)$ for $E \subseteq F$. A Monotone Set Hyperfunction extends this property to subsets of 2^{Ω}. Therefore, the hyperfunction generalizes the monotonicity concept from first-level sets to higher-level structures. □

Iterating this construction, we define monotonicity for functions defined on $P^n(\Omega)$.

Definition 3.31 (Monotone Set n-Superhyperfunction)**.** For $n \geq 1$, a function $\mu : P^n(\Omega) \to \mathbb{R}$ is *monotone* if for all $X, Y \subseteq P^n(\Omega)$ with $X \subseteq Y$, we have:

$$\mu(X) \leq \mu(Y).$$

Example 3.32. Take $\Omega = \{1\}$. Then $P^2(\Omega) = 2^{2^{\Omega}} = \{\emptyset, \{\emptyset\}, \{\{1\}\}, \{\emptyset, \{1\}\}\}$. Define $\mu : P^2(\Omega) \to \mathbb{R}$ by $\mu(Z) = |Z|$. If $X \subseteq Y$ in $P^2(\Omega)$, then $|X| \leq |Y|$, hence $\mu(X) \leq \mu(Y)$, so μ is monotone.

Theorem 3.33. *A Monotone Set n-Superhyperfunction $\mu : P^n(\Omega) \to \mathbb{R}$ is associated with an n-Superhyperstructure.*

Proof. An n-Superhyperstructure is defined on the n-th powerset $P^n(\Omega)$, and a Monotone Set n-Superhyperfunction operates on $P^n(\Omega)$. The monotonicity property ensures that μ respects the hierarchical ordering of subsets within $P^n(\Omega)$. This property directly corresponds to the structural principles of an n-Superhyperstructure, which organizes subsets at multiple levels of powerset iterations. Thus, a Monotone Set n-Superhyperfunction naturally operates within an n-Superhyperstructure. □

Theorem 3.34. *Monotone n-superhyperfunctions generalize monotone set functions and monotone hyperfunctions. Specifically:*

- *For $n = 0$, a monotone 0-superhyperfunction corresponds to a monotone set function $\mu : 2^{\Omega} \to \mathbb{R}$.*
- *For $n = 1$, a monotone 1-superhyperfunction is a monotone hyperfunction $\mu : 2^{2^{\Omega}} \to \mathbb{R}$.*

Proof. The proof follows by inspection of the domain structure. For $n = 0$, $P^0(\Omega) = \Omega$ and monotonicity reduces to the classical definition on subsets of Ω. For $n = 1$, $P^1(\Omega) = 2^{\Omega}$, so monotonicity defined on $P^1(\Omega)$ recovers the monotone hyperfunction definition. For $n > 1$, we obtain the more general n-superhyperfunction case. □

3.4 Modular Hyperfunctions and Their Generalizations

A classical concept closely related to submodularity is *modularity*, which enforces a stricter equality condition [156, 162, 366, 403, 533]. In this subsection, we extend Modular Set Functions to Modular Set Hyperfunctions and Modular Set n-Superhyperfunctions. Relevant definitions and theorems are provided below.

Definition 3.35 (Modular Set Function)**.** (cf. [369, 469]) A set function $\mu : 2^\Omega \to \mathbb{R}$ is *modular* if for all $A, B \subseteq \Omega$:

$$\mu(A \cup B) + \mu(A \cap B) = \mu(A) + \mu(B).$$

Example 3.36. Let $\mu : 2^\Omega \to \mathbb{R}$ be defined by $\mu(S) = c|S|$ for some constant c. For any $A, B \subseteq \Omega$:

$$\mu(A \cup B) + \mu(A \cap B) = c(|A \cup B| + |A \cap B|) = c(|A| + |B|) = \mu(A) + \mu(B).$$

Thus, linear functions of set size are modular.

Theorem 3.37. *A Modular Set Function is a Function.*

Proof. This follows directly from the definition. □

We extend modularity to hyperfunctions.

Definition 3.38 (Modular Set Hyperfunction)**.** A function $\mu : 2^{2^\Omega} \to \mathbb{R}$ is *modular* if for all $A, B \subseteq 2^\Omega$:

$$\mu(A \cup B) + \mu(A \cap B) = \mu(A) + \mu(B).$$

Example 3.39. Consider $\mu : 2^{2^\Omega} \to \mathbb{R}$ defined by $\mu(X) = c|X|$ for some constant c. Using the same argument as for sets:

$$\mu(A \cup B) + \mu(A \cap B) = c(|A \cup B| + |A \cap B|) = c(|A| + |B|) = \mu(A) + \mu(B).$$

Theorem 3.40. *A Modular Set Hyperfunction $\mu : 2^{2^\Omega} \to \mathbb{R}$ is associated with a Hyperstructure.*

Proof. A Hyperstructure is defined on the powerset $\mathcal{P}(\Omega) = 2^\Omega$, and a Modular Set Hyperfunction operates on $\mathcal{P}(\mathcal{P}(\Omega)) = 2^{2^\Omega}$, which is itself a powerset. The modularity property ensures consistency with the union and intersection operations within the powerset. Since the operations $\cup$ and $\cap$ form the basis of the structure of 2^{2^Ω}, the modularity property aligns with the principles of a Hyperstructure. Thus, the Modular Set Hyperfunction operates naturally within a Hyperstructure. □

Theorem 3.41. *A Modular Set Hyperfunction generalizes a Modular Set Function.*

Proof. A Modular Set Function $\mu : 2^\Omega \to \mathbb{R}$ satisfies:

$$\mu(E \cup F) + \mu(E \cap F) = \mu(E) + \mu(F) \quad \text{for all } E, F \subseteq \Omega.$$

A Modular Set Hyperfunction $\mu : 2^{2^\Omega} \to \mathbb{R}$ extends this property to subsets of 2^Ω, operating on the second-level powerset. Hence, a Modular Set Hyperfunction generalizes the modularity property from first-level subsets to second-level subsets. □

Theorem 3.42. *A Modular Set Hyperfunction is a special case of a Submodular Set Hyperfunction.*

Proof. A Modular Set Hyperfunction $\mu : 2^{2^\Omega} \to \mathbb{R}$ satisfies the equality:

$$\mu(A \cup B) + \mu(A \cap B) = \mu(A) + \mu(B), \quad \text{for all } A, B \subseteq 2^\Omega.$$

For a Submodular Set Hyperfunction, the inequality:

$$\mu(A \cup B) + \mu(A \cap B) \leq \mu(A) + \mu(B)$$

holds for all $A, B \subseteq 2^\Omega$. Since equality satisfies the inequality, every Modular Set Hyperfunction is inherently Submodular. However, the converse is not true; there exist Submodular Set Hyperfunctions that do not satisfy the equality condition. □

Definition 3.43 (Modular Set n-Superhyperfunction)**.** A function $\mu : P^n(\Omega) \to \mathbb{R}$ is *modular* if for all $X, Y \subseteq P^n(\Omega)$:

$$\mu(X \cup Y) + \mu(X \cap Y) = \mu(X) + \mu(Y).$$

Example 3.44. For any n, let $\mu : P^n(\Omega) \to \mathbb{R}$ be defined by $\mu(Z) = c|Z|$ for some constant c. Since cardinalities satisfy $|X \cup Y| + |X \cap Y| = |X| + |Y|$ at any set level, μ is modular at all superhyperlevels.

Theorem 3.45. *A Modular Set n-Superhyperfunction $\mu : P^n(\Omega) \to \mathbb{R}$ is associated with an n-Superhyperstructure.*

Proof. An n-Superhyperstructure is defined on the n-th powerset $P^n(\Omega)$. A Modular Set n-Superhyperfunction operates on $P^n(\Omega)$, preserving modularity under the union and intersection operations within $P^n(\Omega)$. The modularity property ensures compatibility with the structural principles of $P^n(\Omega)$, making the function inherently consistent with the n-Superhyperstructure. □

Theorem 3.46. *Modular n-superhyperfunctions generalize modular set functions and modular hyperfunctions. Specifically:*

- *For $n = 0$, a modular 0-superhyperfunction is a modular set function.*
- *For $n = 1$, a modular 1-superhyperfunction is a modular hyperfunction.*

Proof. As in the monotone case, the definitions reduce at lower n. For $n = 0$, $P^0(\Omega) = \Omega$ and sets correspond to 2^Ω. Thus, modularity recovers the classical definition. For $n = 1$, $P^1(\Omega) = 2^\Omega$, so functions on $P^1(\Omega)$ are hyperfunctions. The equality condition for modularity holds at this level as well, giving modular hyperfunctions. Extending to higher n proceeds inductively, preserving the modular equality property at each level. □

Theorem 3.47. *A Modular Set n-Superhyperfunction is a special case of a Submodular Set n-Superhyperfunction.*

Proof. A Modular Set n-Superhyperfunction $\mu : P^n(\Omega) \to \mathbb{R}$ satisfies the equality:

$$\mu(X \cup Y) + \mu(X \cap Y) = \mu(X) + \mu(Y), \quad \text{for all } X, Y \subseteq P^n(\Omega).$$

For a Submodular Set n-Superhyperfunction, the inequality:

$$\mu(X \cup Y) + \mu(X \cap Y) \leq \mu(X) + \mu(Y)$$

holds for all $X, Y \subseteq P^n(\Omega)$. Since equality satisfies the inequality, every Modular Set n-Superhyperfunction is inherently Submodular. However, not all Submodular Set n-Superhyperfunctions satisfy the equality, thus Modular Set n-Superhyperfunctions are a strict subclass of Submodular Set n-Superhyperfunctions. □

3.5 Recursion HyperFunction (HyperRecursion) and Their Generalizations

Recursion is a fundamental concept where the definition of a function, relation, or structure refers back to itself on simpler or previously defined instances [60, 240, 295, 453, 472]. It is widely utilized in programming(cf. [88,269,348,420]), control engineering [266], neural networks [253,255,324,330,506], and machine learning [246, 250]. Here, we extend the notion of recursion from classical functions to hyperfunctions and n-superhyperfunctions, ensuring mathematical rigor and guaranteeing that each recursive definition is well-founded and terminates appropriately.

Definition 3.48 (Recursion Function)**.** (cf. [60, 295, 472]) Let S be a non-empty set. A *Recursion Function* $f : S \to S$ is defined by:

1. A *base case*: A specific element $s_0 \in S$ for which $f(s_0)$ is defined directly and does not depend on f itself.
2. A *recursive step*: For any $s \in S \setminus \{s_0\}$, $f(s)$ is defined in terms of $f(s')$ for some s' that is "simpler" or "previous" according to a well-founded ordering on S.

Formally, suppose $(S, <)$ is a well-founded set. Then a recursion function $f : S \to S$ satisfies:

$$f(s) = \begin{cases} b & \text{if } s = s_0 \text{ (base case)}, \\ \varphi(s, f(s_1), f(s_2), \dots, f(s_k)) & \text{if } s > s_i \text{ for all } i \in \{1, \dots, k\} \text{ and each } s_i < s \text{ (recursive step)}, \end{cases}$$

where φ is a well-defined rule ensuring no infinite descending chain occurs.

Theorem 3.49. *A Recursion Function is a Function.*

Proof. This is evident from the definition. □

Theorem 3.50. *If S is well-founded and a recursion scheme is given by a base case and a recursive step that strictly reduces to simpler cases, then there exists a unique recursion function $f : S \to S$ satisfying the given conditions.*

Proof. This is a standard result in recursion theory. Well-foundedness guarantees that every definition terminates at the base case, while the uniqueness follows from the requirement that each $f(s)$ is determined by previously defined values. □

We now extend recursion to the case of hyperfunctions, where the codomain is a powerset rather than a single set.

Definition 3.51 (Recursion Hyperfunction (HyperRecursion))**.** Let S be a non-empty set and $f : S \to \mathcal{P}(S)$ be a hyperfunction. A *Recursion Hyperfunction* is defined similarly to a recursion function, but:

1. The *base case* specifies $f(s_0) \subseteq S$ directly.
2. The *recursive step* defines $f(s)$ in terms of $f(s_1), f(s_2), \dots, f(s_k)$, ensuring that each s_i is simpler or smaller according to a well-founded order, and that:

$$f(s) = \Psi(s, f(s_1), f(s_2), \dots, f(s_k)) \subseteq S,$$

for some well-defined set-valued function Ψ.

As with recursion functions, the definition must ensure no infinite regress occurs, allowing f to be well-defined and unique under appropriate conditions.

Theorem 3.52. *A Recursion Hyperfunction possesses the structure of a Hyperstructure.*

Proof. A Recursion Hyperfunction is defined as $f : S \to \mathcal{P}(S)$, leveraging a well-founded recursive scheme. A Hyperstructure is built upon $\mathcal{P}(S)$ and involves operations on subsets of S. Since a Recursion Hyperfunction produces elements in $\mathcal{P}(S)$ and can be integrated with hyperoperations defined on $\mathcal{P}(S)$ (e.g., forming weak or strong associative conditions if needed), it inherits the nature of a Hyperstructure. Thus, a Recursion Hyperfunction corresponds to an object that fits naturally into the hyperstructural framework. □

Theorem 3.53. *A Recursion Hyperfunction generalizes a Recursion Function.*

Proof. A Recursion Function $f : S \to S$ maps each element to a single value in S, following a base case and a recursive step. By contrast, a Recursion Hyperfunction $f : S \to \mathcal{P}(S)$ allows each element $s \in S$ to map to a subset of S. If we restrict a Recursion Hyperfunction so that it always returns singleton sets (i.e., each $f(s)$ is a singleton), we recover the behavior of a Recursion Function. Thus, every Recursion Function can be viewed as a special case of a Recursion Hyperfunction. Hence, Recursion Hyperfunctions strictly generalize Recursion Functions. □

Theorem 3.54. *If S is well-founded and a recursion scheme is given for a hyperfunction (with a base case and a strictly reducing recursive step), then there exists a unique recursion hyperfunction $f : S \to \mathcal{P}(S)$ adhering to that scheme.*

Proof. Analogous to the classical case, the well-founded order ensures termination, and each step uniquely determines $f(s)$, guaranteeing existence and uniqueness. □

Finally, we lift the concept to n-superhyperfunctions, where the codomain is the n-th powerset $\mathcal{P}_n(S)$.

Definition 3.55 (Recursion n-Superhyperfunction (Superhyperrecursion))**.** Let S be a non-empty set, and consider an n-superhyperfunction:

$$f : \mathcal{P}_r(S) \to \mathcal{P}_n(S),$$

for some $0 \leq r \leq n$. A *Recursion n-Superhyperfunction* is defined by:

1. A *base case* that directly specifies $f(X_0) \in \mathcal{P}_n(S)$ for a minimal element $X_0 \in \mathcal{P}_r(S)$.
2. A *recursive step* that, for any $X \in \mathcal{P}_r(S)$ greater than the base case element (according to a well-founded order on $\mathcal{P}_r(S)$), defines:

$$f(X) = \Theta(X, f(X_1), f(X_2), \ldots, f(X_k)) \in \mathcal{P}_n(S),$$

where $X_1, \ldots, X_k < X$ and Θ is a suitable set-valued function ensuring a downward chain to the base case.

Theorem 3.56. *A Recursion n-Superhyperfunction generalizes both Recursion Hyperfunctions and Recursion Functions.*

Proof. A Recursion n-Superhyperfunction $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ operates at the n-th powerset level. For $n = 1$ and $r = 0$, this reduces to a Recursion Hyperfunction $f : S \to \mathcal{P}(S)$. By further restricting the image to singletons, we recover a classical Recursion Function. Thus, the Recursion n-Superhyperfunction framework includes both Recursion Hyperfunctions (when $n = 1$) and Recursion Functions (when restricting the image to singletons). Therefore, Recursion n-Superhyperfunctions generalize both Recursion Hyperfunctions and Recursion Functions. □

Theorem 3.57. *A Recursion n-Superhyperfunction possesses the structure of an n-Superhyperstructure.*

Proof. By definition, an n-Superhyperfunction $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ maps elements of $\mathcal{P}_r(S)$ into $\mathcal{P}_n(S)$, where $\mathcal{P}_n(S)$ is the n-th powerset of S. An n-Superhyperstructure is characterized by operations defined on $\mathcal{P}_n(S)$. Since the recursion scheme on $\mathcal{P}_n(S)$ ensures a well-founded order and defines $f(X)$ recursively in terms of previously defined values at lower elements, f inherently respects the layered structure of the n-th powerset. This aligns with the hierarchical nature of an n-Superhyperstructure. Consequently, a Recursion n-Superhyperfunction, defined via a recursion process on $\mathcal{P}_n(S)$, operates naturally within the n-Superhyperstructural framework. □

Theorem 3.58. *If $\mathcal{P}_r(S)$ is equipped with a well-founded order and the recursion scheme for the n-superhyperfunction is well-defined (having a proper base case and a strictly reducing recursive step), then there exists a unique recursion n-superhyperfunction $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$.*

Proof. The argument extends the previous proofs. The well-founded order ensures no infinite descending sequences, and each definition step determines $f(X)$ uniquely. Thus, existence and uniqueness follow by the same reasoning as for recursion functions and recursion hyperfunctions. □

3.6 Random HyperFunction and Their Generalizations

In this subsection, we discuss the concept of Random HyperFunction.

3.6.1 Random HyperFunction and n-SuperHyperFunction

Random function means a function defined over a probability space, producing unpredictable outputs, used in programming [202, 332, 576], probability [130, 321, 425], neural networks [373, 561], graphs [58, 143].

Definition 3.59 (Random Function)**.** (cf. [230]) Let S be a non-empty set and let Ω be a probability space. A *Random Function* is a mapping:

$$F : \Omega \times S \to S$$

such that for each fixed $\omega \in \Omega$, the mapping $F(\omega, \cdot) : S \to S$ is a function. In other words, a random function assigns to each input $x \in S$ an output $F(\omega, x) \in S$, depending on a random parameter ω. This construction allows uncertainty or variability in the function's values.

Example 3.60. Let $S = \{1, 2, 3\}$ and $\Omega = [0, 1]$ with a uniform probability measure. Define:

$$F(\omega, x) = \begin{cases} 1 & \text{if } \omega < 1/3, \\ 2 & \text{if } 1/3 \le \omega < 2/3, \\ 3 & \text{if } \omega \ge 2/3. \end{cases}$$

For each ω, $F(\omega, \cdot)$ is a function that maps every element of S to a single element of S (here it's a trivial function that doesn't even depend on x). This is a random function because for different ω values we might get different constant outputs.

Theorem 3.61. *A Random Function is a Function.*

Proof. This follows directly from the definition. □

Definition 3.62 (Random Hyperfunction)**.** Let S be a non-empty set, and Ω a probability space. A *Random Hyperfunction* is a mapping:

$$F : \Omega \times S \to \mathcal{P}(S)$$

such that for each fixed $\omega \in \Omega$, the mapping $F(\omega, \cdot) : S \to \mathcal{P}(S)$ is a hyperfunction. That is, for each ω and each $x \in S$, $F(\omega, x) \subseteq S$. This allows for both randomness (from ω) and multi-valuedness (from the hyperfunction nature).

Example 3.63. Let $S = \{a, b\}$ and $\Omega = [0, 1]$ with a uniform probability measure. Define:

$$F(\omega, x) = \begin{cases} \{a\} & \text{if } \omega < 1/2, \\ \{a, b\} & \text{if } \omega \ge 1/2. \end{cases}$$

Here, for each ω, $F(\omega, \cdot)$ is a hyperfunction since it maps an element $x \in S$ to a subset of S. For $\omega < 1/2$, $F(\omega, x) = \{a\}$ for any x. For $\omega \ge 1/2$, $F(\omega, x) = \{a, b\}$ for any x. Thus, this is a random hyperfunction.

Theorem 3.64. *A Random Hyperfunction generalizes a Random Function.*

Proof. A Random Function $F : \Omega \times S \to S$ assigns a single element of S to each pair (ω, x). A Random Hyperfunction $F : \Omega \times S \to \mathcal{P}(S)$ assigns a subset of S to each pair (ω, x). If we restrict a Random Hyperfunction to always produce singleton subsets, we recover the behavior of a Random Function. Thus, every Random Function is a special case of a Random Hyperfunction, proving that Random Hyperfunctions strictly generalize Random Functions. □

Theorem 3.65. *A Random Hyperfunction inherits the structure of a Hyperstructure.*

Proof. A Hyperstructure is built on $\mathcal{P}(S)$. For a Random Hyperfunction, each fixed ω yields a hyperfunction $F(\omega, \cdot) : S \to \mathcal{P}(S)$. Since hyperfunctions naturally operate within the framework of Hyperstructures, the random parameter ω does not disrupt this structural alignment. Each ω-slice of the Random Hyperfunction is a Hyperfunction, hence associated with a Hyperstructure. Thus, a Random Hyperfunction aligns with the structure of a Hyperstructure at each fixed ω. □

Definition 3.66 (Random n-Superhyperfunction)**.** Let S be a non-empty set, Ω a probability space, and let $\mathcal{P}_n(S)$ denote the n-th powerset of S. A *Random n-Superhyperfunction* is a mapping:

$$F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$$

for some $0 \leq r \leq n$, such that for each fixed $\omega \in \Omega$, $F(\omega, \cdot) : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ is an n-Superhyperfunction. Thus, we have both the iterative powerset structure from the n-superhyperfunction and the randomness from the probability space.

Example 3.67. Consider $S = \{x\}$. Then $\mathcal{P}(S) = \{\emptyset, \{x\}\}$ and $\mathcal{P}_2(S) = \mathcal{P}(\mathcal{P}(S)) = \{\emptyset, \{\emptyset\}, \{\{x\}\}, \{\emptyset, \{x\}\}\}$. Let $\Omega = [0, 1]$ with a uniform measure. Define:

$$F(\omega, X) = \begin{cases} \{\{x\}\} & \text{if } \omega < 1/3, \\ \{\emptyset, \{x\}\} & \text{if } 1/3 \leq \omega < 2/3, \\ \{\emptyset\} & \text{if } \omega \geq 2/3. \end{cases}$$

For each fixed ω, $F(\omega, \cdot)$ maps elements of $\mathcal{P}(S)$ (which is $\{\emptyset, \{x\}\}$ at the base level if $r = 1$ or more complicated sets if $r > 1$) to subsets of $\mathcal{P}_2(S)$. This is a random n-superhyperfunction for $n = 2$ (and an appropriate choice of r) that incorporates both multiple levels of powersets and randomness.

Theorem 3.68. *A Random n-Superhyperfunction generalizes a Random Hyperfunction (and hence also a Random Function).*

Proof. A Random n-Superhyperfunction $F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ reduces to a Random Hyperfunction when $n = 1$ and $r = 0$, because $\mathcal{P}_1(S) = \mathcal{P}(S)$. Since a Random Hyperfunction generalizes a Random Function, it follows by transitivity that a Random n-Superhyperfunction generalizes both Random Hyperfunctions and Random Functions. In other words, by setting $n = 1$ and restricting images to singletons, we can emulate a Random Function, and by just setting $n = 1$ without further restrictions, we emulate a Random Hyperfunction. Hence, the n-Superhyperfunction framework is strictly more general. □

Theorem 3.69. *A Random n-Superhyperfunction inherits the structure of an n-Superhyperstructure.*

Proof. For an n-Superhyperfunction $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$, we know it is associated with an n-Superhyperstructure. Introducing randomness via Ω to form a Random n-Superhyperfunction $F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ means that for each ω, $F(\omega, \cdot)$ is an n-Superhyperfunction. Since each n-Superhyperfunction corresponds to an n-Superhyperstructure, the random extension preserves the hierarchical structure. Hence, a Random n-Superhyperfunction aligns with the structure of an n-Superhyperstructure. □

3.6.2 Hyper-randomness and Their Generalizations

Randomness refers to the lack of pattern or predictability in events, commonly modeled in probability and statistics [78, 101, 218, 555]. HyperRandomness describes deviations from classical randomness, exhibiting non-stationary, dependent, and unstable statistical properties.

Definition 3.70 (Randomness)**.** [78,101,218,555] Let $\{X_i\}_{i=1}^{\infty}$ be a sequence of random variables defined on a probability space $(\Omega, \mathcal{F}, P)$. We say the sequence exhibits *Randomness* if:

1. The random variables X_i are independent: for any finite collection $X_{i_1}, X_{i_2}, \ldots, X_{i_k}$, their joint distribution factors into the product of their marginal distributions.
2. The random variables X_i are identically distributed (IID): each X_i follows the same probability distribution.

In essence, randomness here refers to a scenario where the data exhibits no temporal correlation and stable statistical properties, commonly seen in IID models such as radioactive decay counts or coin flips.

Example 3.71. Consider a sequence $\{X_i\}$ where each X_i is a Bernoulli random variable with parameter $p = 1/2$. Each trial is independent and identically distributed. This sequence is random in the classical sense.

Theorem 3.72. *If a Random Function $F : \Omega \times S \to S$ is endowed with a hyperoperation defined on its range, it can be embedded into a Hyperstructure.*

Proof. A Random Function F produces single-valued outputs for each pair (ω, x). Consider a hyperoperation $\circ$ defined on S that, for any two elements, produces a subset of S. If we incorporate this hyperoperation into the image of F, we can view the image sets under different ω as collections of subsets (albeit trivial singletons). By suitably extending or refining $\circ$ to allow non-trivial set outputs, we can interpret the resulting structure as a Hyperstructure. Thus, even though the original function is single-valued, the introduction of a hyperoperation on its codomain allows embedding into a Hyperstructure framework. □

Theorem 3.73. *Consider a Random Hyperfunction $F : \Omega \times S \to \mathcal{P}(S)$ that exhibits classical randomness (IID conditions) at each ω-slice. Then it can be simplified to a Random Function while maintaining structural properties.*

Proof. If the Random Hyperfunction F always yields subsets that are, with high probability, singletons due to IID conditions and stationarity, then for almost all ω, $F(\omega, \cdot)$ behaves like a function into a single element. By imposing these constraints, we effectively collapse the hyperfunction into a single-valued function. This shows that Random Hyperfunctions can be restricted back to Random Functions while retaining the idea of randomness. The result emphasizes that Random Functions are special cases of Random Hyperfunctions. □

Definition 3.74 (Hyperrandomness)**.** (cf. [204,205,582]) *Hyperrandomness* refers to a phenomenon where the statistical properties of the data change over time or across subsets, violating the IID assumptions. Specifically, a sequence $\{X_i\}$ exhibiting hyperrandomness has:

1. Non-independence: There is temporal or structural dependence among the X_i.
2. Non-stationarity: The distribution of X_i changes over time (e.g., means, variances evolve over time).
3. Statistical instability: Metrics like cumulative averages and cumulative variances may not stabilize and may increase with sample size N.

Hyperrandomness captures scenarios where external factors or underlying processes cause departures from classical randomness. This leads to complex and evolving statistical behaviors that cannot be described by fixed distributions.

Example 3.75. Suppose $\{X_i\}$ is a sequence of real-valued observations where:

- For $i \leq N_0$, $X_i \sim N(0, 1)$ (a normal distribution with mean 0, variance 1).
- For $i > N_0$, the distribution gradually shifts to $X_i \sim N(\mu_i, \sigma_i^2)$ with μ_i increasing over time.

The resulting sequence is non-stationary and may exhibit autocorrelation, making it hyperrandom.

Theorem 3.76. *Hyperrandomness generalizes Randomness.*

Proof. If we impose IID conditions and stationarity on a hyperrandom process, we return to a classical random process. Hence, randomness is a special case of hyperrandomness where non-independence and non-stationarity vanish, and statistical properties stabilize. Thus, hyperrandomness is strictly more general, encompassing randomness as a subset scenario. □

Theorem 3.77. *Hyperrandomness can be modeled within a Hyperstructure framework.*

Proof. Hyperrandomness involves complex, time-evolving statistical behaviors. Consider representing the temporal sequences of distributions as elements in a set S and defining a hyperoperation $\circ$ that, when applied to pairs of temporal windows or subsets, yields sets of possible future distributions (reflecting uncertainty and instability). This construction yields a Hyperstructure $(S, \circ)$ capturing hyperrandomness. The sets generated by $\circ$ can represent evolving distributions and statistical properties. Thus, hyperrandomness can be described using a hyperstructural approach. □

Definition 3.78 (n-Superhyperrandomness)**.** *n-Superhyperrandomness* generalizes hyperrandomness by considering multiple iterative layers of complexity and hierarchical structure. At each level, the distributional properties and dependencies can evolve, creating a multi-layered non-stationary and non-IID scenario.

Formally, consider a sequence $\{X_i^{(n)}\}$ defined on $\mathcal{P}_n(S)$ or representing n-th order complexities. n-Superhyperrandomness occurs if at each of the n layers, the statistical properties can change, show non-independence, non-stationarity, and statistical instability, extending the concept of hyperrandomness to an n-th power set level of complexity.

Example 3.79. Imagine a scenario with $n = 2$. At the first level, you have a sequence of sets $\{A_i\} \subseteq \mathcal{P}(S)$ whose structure evolves over time. At the second level, you have $\{X_i^{(2)}\} \in \mathcal{P}_2(S)$ capturing even more complex patterns (sets of sets), with distributions changing over multiple hierarchical layers. Variances and means defined at each hierarchical level fail to stabilize, revealing n-superhyperrandomness.

Theorem 3.80. *n-Superhyperrandomness generalizes Hyperrandomness (and thus Randomness).*

Proof. For $n = 1$, n-superhyperrandomness reduces to hyperrandomness. Since hyperrandomness reduces to randomness under additional constraints (IID and stationarity), it follows that n-superhyperrandomness generalizes both hyperrandomness and randomness. With increasing n, we add layers of complexity and structural changes, making n-superhyperrandomness strictly more general. □

Definition 3.81 (Hyperrandomness and Random Hyperfunctions)**.** A Random Hyperfunction $F : \Omega \times S \to \mathcal{P}(S)$ is said to exhibit hyperrandomness if the underlying random parameters and outputs at the powerset level show non-stationary, dependent, and unstable statistical behaviors over time or across indices.

Theorem 3.82. *A Random Hyperfunction with hyperrandom behavior generalizes a Random Function with random behavior.*

Proof. If the Random Hyperfunction simplifies to always produce singleton sets and reverts to IID and stationary distributions, it becomes a Random Function with classical randomness. Allowing sets of outputs and dropping IID and stationarity leads to hyperrandomness. Thus, hyperrandomness combined with hyperfunctionality generalizes the classical notion of randomness in functions. □

Definition 3.83 (n-Superhyperrandomness and Random n-Superhyperfunction)**.** A Random n-Superhyperfunction $F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ exhibits n-superhyperrandomness if at each powerset iteration level, we have non-stationary, dependent, and statistically unstable behaviors extending through multiple layers.

Theorem 3.84. *A Random n-Superhyperfunction exhibiting n-superhyperrandomness generalizes all previously defined notions (Random Functions, Random Hyperfunctions, Hyperrandomness) and aligns with n-Superhyperstructures.*

Proof. A Random n-Superhyperfunction is the most general construct discussed, incorporating randomness, multi-level sets, and complex dependence structures. When it exhibits n-superhyperrandomness, it encompasses classical randomness (if constraints are applied), hyperrandomness (if reduced to fewer layers), and random hyperfunctionality. Additionally, since n-Superhyperfunctions naturally correspond to n-Superhyperstructures, adding n-superhyperrandomness aligns this construct with the complexity and hierarchical definitions of n-Superhyperstructures. Thus, it generalizes all previously considered notions. □

Theorem 3.85. *Let $F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ be an n-Superhyperfunction. If F exhibits only classical randomness (IID and stationary conditions) at all n levels, then F reduces to a Random Hyperfunction and further to a Random Function by successive restrictions.*

Proof. Since an n-Superhyperfunction generalizes a Hyperfunction, and a Hyperfunction generalizes a Function, the presence of classical randomness at all levels ensures that the complexity introduced by iterative powersets and multi-level sets does not require the extra degrees of freedom. Hence, by constraining each level to singletons and stable distributions, we revert through each hierarchical layer back to a Random Hyperfunction and eventually to a Random Function. This telescoping argument demonstrates that n-Superhyperfunctions with restrictive conditions can mimic simpler structures. □

Theorem 3.86. *n-Superhyperrandomness can be modeled within an n-Superhyperstructure framework.*

Proof. Consider the n-th powerset $\mathcal{P}_n(S)$. Elements of $\mathcal{P}_n(S)$ represent hierarchical, iterative subsets of S. In n-superhyperrandomness, we have time-evolving, non-stationary, and dependent distributions associated with these elements. We need to encode both the hierarchical structure and the evolving statistical properties into a suitable algebraic setting.

Let $\circ_n$ be a hyperoperation on $\mathcal{P}_n(S)$ that takes two n-th level subsets and produces a set of possible future subsets that reflect the changing distributions and dependencies. For example, if $X, Y \in \mathcal{P}_n(S)$ represent current statistical states (e.g., distributions, cumulative averages) at the n-th level, define:

$$X \circ_n Y = \{Z \in \mathcal{P}_n(S) : Z \text{ is compatible with the combined statistical evolution of } X \text{ and } Y\}.$$

This operation produces a set of potential outcomes, each corresponding to a distinct statistical trajectory. The non-uniqueness and complexity reflect n-superhyperrandomness.

An n-Superhyperstructure is defined on $\mathcal{P}_n(S)$ with a hyperoperation. By defining $\circ_n$ in a way that encodes non-stationarity, non-independence, and statistical instability, we ensure that the resulting structure $(\mathcal{P}_n(S), \circ_n)$ captures the essence of n-superhyperrandomness. The hyperoperation $\circ_n$ does not produce a single outcome but a set of outcomes, allowing for flexible modeling of uncertainty and complexity.

Since $(\mathcal{P}_n(S), \circ_n)$ is by construction an n-Superhyperstructure, and we have shown how to incorporate n-superhyperrandomness into its definition, it follows that n-Superhyperrandomness can indeed be represented within an n-Superhyperstructure framework.

Thus, Theorem 3.86 is proved. □

Theorem 3.87. *A Random n-Superhyperfunction exhibiting n-superhyperrandomness can be naturally integrated into an n-Superhyperstructure that models its complexity and non-stationary statistical properties.*

Proof. A Random n-Superhyperfunction $F : \Omega \times \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ provides the randomness and hierarchical set structure. Adding n-superhyperrandomness means that the statistical properties at each level evolve over time. According to Theorem 3.86, this evolution can be encoded into the hyperoperation of an n-Superhyperstructure. Thus, by selecting an appropriate hyperoperation $\circ_n$, we represent both the n-superhyperfunction's hierarchical mapping and the n-superhyperrandomness' complexity. As a result, the entire phenomenon is captured within an n-Superhyperstructure, confirming the alignment of these concepts. □

4 Future Directions of this Paper: Extensions Using Superhyperstructures

This section outlines the future directions of this research. The primary focus is to explore the applicability of Hyperstructures and SuperHyperstructures to other concepts and to investigate potential applications stemming from these extensions. Additionally, future work will focus on the development of construction and recognition algorithms tailored to the Hyperstructure and SuperHyperstructure frameworks.

4.1 Future Directions: Weak Hyperstructures and n-Weak Superhyperstructures

Several studies have explored Weak Hyperstructures as extensions of the broader Hyperstructure framework [15, 18, 119, 139, 278, 280, 376, 539, 543]. As noted in the introduction, Hyperstructures are commonly used in group theory and algebra theory. However, this paper focuses on their examination in the context of power sets, a distinction readers should note.

Building on this foundation, the subsection proposes extending these concepts further to n-Weak Superhyperstructures. It is anticipated that future research will expand these ideas to encompass applications in graph theory, networks, and algebraic systems. It is important to note that, depending on the context of a given study, Hyperstructure research may overlap with algebraic studies. Therefore, caution is advised when interpreting the scope and definitions used in different papers.

Definition 4.1 (Weak Hyperstructure)**.** (cf. [18, 119, 139, 543]) A *Weak Hyperstructure* is a generalization of a Hyperstructure in which the associative axiom is replaced by a weaker condition. Let S be a non-empty set, and let $\mathcal{P}(S)$ denote the powerset of S. A Weak Hyperstructure is a pair:

$$\mathcal{W} = (\mathcal{P}(S), \circ),$$

where $\circ : \mathcal{P}(S) \times \mathcal{P}(S) \to \mathcal{P}(S)$ is a hyperoperation satisfying the following *weak associativity* condition:

$$\forall A, B, C \in \mathcal{P}(S), \quad \left(\bigcup_{x \in A \circ (B \circ C)} x \right) \cap \left(\bigcup_{y \in (A \circ B) \circ C} y \right) \neq \emptyset.$$

In other words, while a strict Hyperstructure requires $A \circ (B \circ C) = (A \circ B) \circ C$, a Weak Hyperstructure only demands that the results of these operations share at least one common element.

Theorem 4.2. *Every Hyperstructure is a Weak Hyperstructure, but not every Weak Hyperstructure is a Hyperstructure.*

Proof. If $(\mathcal{P}(S), \circ)$ is a Hyperstructure, then associativity holds exactly:

$$A \circ (B \circ C) = (A \circ B) \circ C.$$

This implies that:

$$\bigcup_{x \in A \circ (B \circ C)} x = \bigcup_{y \in (A \circ B) \circ C} y,$$

hence the weak condition (non-empty intersection) is trivially satisfied.

Conversely, a Weak Hyperstructure may satisfy only the weaker condition. Such a structure need not have strict associativity, allowing counterexamples where the two sides do not match exactly but still have a non-empty intersection. Thus, not all Weak Hyperstructures are Hyperstructures. □

Definition 4.3 (n-Weak Superhyperstructure)**.** For $n \geq 1$, an *n-Weak Superhyperstructure* is defined by iterating the powerset construction n-times. Let $\mathcal{P}^0(S) = S$ and $\mathcal{P}^{k+1}(S) = \mathcal{P}(\mathcal{P}^k(S))$.

An n-Weak Superhyperstructure is:

$$\mathcal{WSH}_n = (\mathcal{P}_n(S), \circ),$$

where $\circ : \mathcal{P}_n(S) \times \mathcal{P}_n(S) \to \mathcal{P}_n(S)$ satisfies the weak associativity condition at the n-th level of iteration:

$$\forall A, B, C \in \mathcal{P}_n(S), \quad \left(\bigcup_{x \in A \circ (B \circ C)} x \right) \cap \left(\bigcup_{y \in (A \circ B) \circ C} y \right) \neq \emptyset.$$

Theorem 4.4. *Every n-Superhyperstructure is an n-Weak Superhyperstructure, but not every n-Weak Superhyperstructure is an n-Superhyperstructure.*

Proof. If strict associativity holds at the n-th level, it trivially implies the weak associativity condition. However, a weak condition does not guarantee equality, just a non-empty intersection. Hence the inclusion. □

4.1.1 Weak Hypergraph

The authors believe that Weak Hyperstructures have the potential for various applications across multiple domains. As an example, we consider the field of graph theory.

Hypergraphs serve as a fundamental example of Hyperstructures [59,75]. By analogy, this paper introduces the concepts of Weak Hypergraphs and n-Weak Superhypergraphs, extending the traditional hypergraph framework by incorporating the principle of weak associativity into their structural definitions. It is important to note that this paper omits discussions on the basic principles of graph theory. For a comprehensive understanding of graph theory and its notations, readers are encouraged to consult [134–136, 211, 213, 326].

Definition 4.5 (Graph)**.** [136] A *graph* G is a mathematical structure used to represent relationships between objects. It comprises a set of vertices $V(G)$ and a set of edges $E(G)$, where each edge connects a pair of vertices. Formally, a graph is represented as $G = (V, E)$, where V is the set of vertices and E is the set of edges.

Definition 4.6 (Hypergraph)**.** [59] A *hypergraph* $H = (V, E)$ consists of a set V of vertices and a set E of hyperedges. Each hyperedge $e \in E$ is a subset of V, meaning $e \subseteq V$ and $E \subseteq \mathcal{P}(V)$, where $\mathcal{P}(V)$ denotes the power set of V.

Example 4.7 (Hypergraph)**.** [59] Let $V = \{1, 2, 3\}$ be a set of vertices. Consider the hyperedges:

$$E = \{\{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}.$$

Then $H = (V, E)$ is a hypergraph. Here:

- Each hyperedge is simply a subset of V.
- There is no operation defined on the set of hyperedges, so no notion of associativity arises.

This is a standard hypergraph: it lists a collection of subsets of V as hyperedges, but does not impose any additional structure.

Definition 4.8 (Weak Hypergraph)**.** A *Weak Hypergraph* is a triple $(V, E, \circ)$ where:

- V is a non-empty set of vertices.
- $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ is a collection of hyperedges.
- $\circ : E \times E \to \mathcal{P}(E)$ is a hyperoperation on the hyperedges.

The weak associativity condition for a Weak Hypergraph is:

$$\forall A, B, C \in E, \quad \left(\bigcup_{x \in A \circ (B \circ C)} x \right) \cap \left(\bigcup_{y \in (A \circ B) \circ C} y \right) \neq \emptyset.$$

This structure does not require strict associativity of hyperedges but only that the outcomes have some overlap.

Now we introduce a hyperoperation $\circ$ on the set of hyperedges to form a Weak Hypergraph. The key difference is that we will not require strict associativity. Instead, we only need that any two different ways of associating the hyperoperation produce results with a non-empty intersection.

Example 4.9 (Weak Hypergraph)**.** Let $V = \{a, b\}$ be a vertex set, and let the hyperedges be:

$$E = \{A, B, C\} \quad \text{where} \quad A = \{a\}, B = \{b\}, C = \{a, b\}.$$

Define a hyperoperation $\circ : E \times E \to \mathcal{P}(E)$ as follows:

$$A \circ B = \{A, C\}, \quad B \circ C = \{C\}, \quad A \circ C = \{B, C\}, \quad C \circ A = \{A, B\}, \quad C \circ B = \{B\}, \quad B \circ A = \{C\}.$$

Check weak associativity for a triple, say (A, B, C):

$$A \circ (B \circ C) = A \circ \{C\} = A \circ C = \{B, C\},$$

while

$$(A \circ B) \circ C = \{A, C\} \circ C.$$

Now $\{A, C\} \circ C = \bigcup_{x \in \{A,C\}} x \circ C$. Since $A \circ C = \{B, C\}$ and $C \circ C$ can be defined similarly (suppose $C \circ C = \{C\}$), then:

$$(A \circ B) \circ C = (\{A, C\} \circ C) = \{B, C\} \cup \{C\} = \{B, C\}.$$

We have:

$$A \circ (B \circ C) = \{B, C\} \quad \text{and} \quad (A \circ B) \circ C = \{B, C\}.$$

In this case, they are actually equal, but even if we had a scenario where they differ but share a non-empty intersection, the structure would still be a Weak Hypergraph. The critical difference from a standard Hypergraph is the presence of the operation $\circ$ and the relaxed (weak) associativity condition.

Example 4.10 (Distinguishing Hypergraph and Weak Hypergraph)**.** Consider a hypergraph $H = (V, E)$ with:

$$V = \{x, y\}, \quad E = \{\{x\}, \{y\}\}.$$

As a plain hypergraph, no operation is defined on E. We just have two hyperedges $\{x\}$ and $\{y\}$.

Now suppose we define an operation $\circ$ on E by:

$$\{x\} \circ \{x\} = \{\{x\}\}, \quad \{y\} \circ \{y\} = \{\{y\}\}, \quad \text{and} \quad \{x\} \circ \{y\} = \{\{x\}, \{y\}\}, \quad \{y\} \circ \{x\} = \{\{x\}\}.$$

Check associativity:

$$(\{x\} \circ \{x\}) \circ \{y\} = \{\{x\}\} \circ \{y\} = \{\{x\}, \{y\}\},$$

while

$$\{x\} \circ (\{x\} \circ \{y\}) = \{x\} \circ \{\{x\}, \{y\}\}.$$

Since $\{x\} \circ \{y\}$ yields $\{\{x\}, \{y\}\}$, applying $\{x\}$ to each element in $\{\{x\}, \{y\}\}$ might produce a different set, and you might not get equality. However, if you ensure they share at least one common element, you have a Weak Hypergraph. If equality fails but intersection is non-empty, associativity is not strict. This difference shows that adding the operation $\circ$ and relaxing associativity transforms a standard Hypergraph into a Weak Hypergraph.

Theorem 4.11. *Every Hypergraph can be viewed as a Weak Hypergraph, but not every Weak Hypergraph corresponds to a strictly associative Hypergraph structure.*

Proof. Follows directly from the analogy with Hyperstructures. If associativity of hyperedges is strict, it trivially satisfies the weak condition. If only the weak condition holds, the structure is weaker and does not imply strict equality of outcomes. □

Theorem 4.12. *A Weak Hypergraph possesses the structure of a Weak Hyperstructure.*

Proof. A Weak Hypergraph $(V, E, \circ)$ is defined with:

- V, a set of vertices.
- $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$, a collection of hyperedges.

- $\circ : E \times E \to \mathcal{P}(E)$, a hyperoperation satisfying weak associativity:

$$\forall A, B, C \in E, \quad \left(\bigcup_{x \in A \circ (B \circ C)} x \right) \cap \left(\bigcup_{y \in (A \circ B) \circ C} y \right) \neq \emptyset.$$

The pair $(\mathcal{P}(E), \circ)$ forms a Weak Hyperstructure because $\circ$ satisfies weak associativity and the domain $\mathcal{P}(E)$ aligns with the structure of a Weak Hyperstructure. Thus, a Weak Hypergraph is a specific instance of a Weak Hyperstructure. □

Definition 4.13 (n-SuperHyperGraph)**.** (cf. [184, 487, 488]) Let V_0 be a finite set of base vertices. Define the n-th iterated power set of V_0 recursively as:

$$\mathcal{P}^0(V_0) = V_0, \quad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\left(\mathcal{P}^k(V_0)\right),$$

where $\mathcal{P}(A)$ denotes the power set of set A.

An *n-SuperHyperGraph* is an ordered pair $H = (V, E)$, where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*, which are elements of the n-th power set of V_0.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of *superedges*, also elements of $\mathcal{P}^n(V_0)$.

Each supervertex $v \in V$ can be:

- A single vertex ($v \in V_0$),
- A subset of V_0 ($v \subseteq V_0$),
- A subset of subsets of V_0, up to n levels ($v \in \mathcal{P}^n(V_0)$),
- An indeterminate or fuzzy set(cf. [568]),
- The null set ($v = \emptyset$).

Each superedge $e \in E$ connects supervertices, potentially at different hierarchical levels up to n.

Definition 4.14 (n-Weak Superhypergraph)**.** An *n-Weak Superhypergraph* generalizes the concept of a Weak Hypergraph by iterating the powerset operation n-times on the edge sets. Formally, let $\mathcal{P}^n(V)$ be the n-th powerset of V. Define:

$$\mathcal{WGH}_n = (V, \mathcal{E}_n, \circ),$$

where $\mathcal{E}_n \subseteq \mathcal{P}^n(V)$ and $\circ : \mathcal{E}_n \times \mathcal{E}_n \to \mathcal{P}(\mathcal{E}_n)$ satisfies the weak associativity condition at the n-th level.

Example 4.15 (n-Weak Superhypergraph)**.** Let $V = \{u\}$. Then $\mathcal{P}(V) = \{\emptyset, \{u\}\}$. For $n = 2$, $\mathcal{P}_2(V) = \mathcal{P}(\mathcal{P}(V)) = \{\emptyset, \{\emptyset\}, \{\{u\}\}, \{\emptyset, \{u\}\}\}$.

Define $\mathcal{E}_2 \subseteq \mathcal{P}_2(V)$ to be a small subset, for example:

$$\mathcal{E}_2 = \{\{\emptyset\}, \{\{u\}\}, \{\emptyset, \{u\}\}\}.$$

Then define a hyperoperation $\circ : \mathcal{E}_2 \times \mathcal{E}_2 \to \mathcal{P}(\mathcal{E}_2)$ that satisfies the weak associativity condition at this second level of iteration (i.e., ensure that for any triple of elements from $\mathcal{E}_2$, the two different ways of composing them share a non-empty intersection). Without detailing a full table, one can choose $\circ$ so that it is not strictly associative, but still ensures that each pairwise composition leads to overlapping results.

This construction yields an n-Weak Superhypergraph (with $n = 2$ in this example), generalizing the Weak Hypergraph concept to higher-order powersets.

Theorem 4.16. *Every n-Superhypergraph is an n-Weak Superhypergraph, but the converse is not necessarily true.*

Proof. This result is analogous to the relationship between n-Superhyperstructures and n-Weak Superhyperstructures. Strict associativity at the n-th level implies the weak condition, but the weak condition alone does not guarantee strict associativity. □

Theorem 4.17. *An n-Weak Superhypergraph possesses the structure of an n-Weak Superhyperstructure.*

Proof. An n-Weak Superhypergraph $\mathcal{WGH}_n = (V, \mathcal{E}_n, \circ)$ is defined with:

- V, a set of vertices.
- $\mathcal{E}_n \subseteq \mathcal{P}^n(V)$, a set of n-th level hyperedges.
- $\circ : \mathcal{E}_n \times \mathcal{E}_n \to \mathcal{P}(\mathcal{E}_n)$, satisfying weak associativity at the n-th level:

$$\forall A, B, C \in \mathcal{E}_n, \quad \left(\bigcup_{x \in A \circ (B \circ C)} x \right) \cap \left(\bigcup_{y \in (A \circ B) \circ C} y \right) \neq \emptyset.$$

The pair $(\mathcal{P}_n(V), \circ)$ forms an n-Weak Superhyperstructure because $\mathcal{P}_n(V)$ is the domain and $\circ$ satisfies weak associativity. Therefore, n-Weak Superhypergraphs are examples of n-Weak Superhyperstructures. □

4.1.2 Weak Hyperfunction

In this subsubsection, we provide a more mathematically precise and detailed formulation of Weak Hyperfunctions and Weak n-Superhyperfunctions. Although these concepts are still in the conceptual stage, the definitions and theorems below aim to clarify their structure and properties. Future work may refine these notions further.

Notation 4.18 (Hyperfunction Composition (Recall))**.** *Let S be a non-empty set, and let $\mathcal{P}(S)$ denote the powerset of S. A* hyperfunction *is a function $f : S \to \mathcal{P}(S)$. Given two hyperfunctions $f, g : S \to \mathcal{P}(S)$, their composition $f \circ g : S \to \mathcal{P}(S)$ is defined as:*

$$(f \circ g)(x) = \bigcup_{y \in g(x)} f(y).$$

This definition ensures that the image of x under $f \circ g$ is the union of the images under f of all elements in the image $g(x)$.

For three hyperfunctions $f, g, h : S \to \mathcal{P}(S)$, we can consider their compositions in two different ways:

$$f \circ (g \circ h) \quad \text{and} \quad (f \circ g) \circ h.$$

Strict associativity would require these two compositions to produce exactly the same set for every $x \in S$.

Weak Hyperfunctions relax the condition of strict associativity. Instead of requiring equality, we only demand that the results of different parenthesizations have a non-empty intersection for every input.

Definition 4.19 (Weak Hyperfunction)**.** Let $\mathcal{F}$ be a collection of hyperfunctions $f : S \to \mathcal{P}(S)$. We say that $\mathcal{F}$ consists of *Weak Hyperfunctions* if for every triple $(f, g, h) \in \mathcal{F}^3$ and for every $x \in S$,

$$(f \circ (g \circ h))(x) \cap ((f \circ g) \circ h)(x) \neq \emptyset.$$

In other words, the triple (f, g, h) is *weakly associative* if no matter how we parenthesize the compositions, the resulting sets have at least one common element at each $x \in S$.

This condition is strictly weaker than associativity. Associativity demands:

$$f \circ (g \circ h) = (f \circ g) \circ h,$$

while weak associativity only insists on:

$$(f \circ (g \circ h))(x) \cap ((f \circ g) \circ h)(x) \neq \emptyset \quad \text{for all } x \in S.$$

Theorem 4.20. *If $f, g, h : S \to \mathcal{P}(S)$ are strictly associative, then they are also weakly associative.*

Proof. Strict associativity implies equality of sets. Thus, if $f \circ (g \circ h) = (f \circ g) \circ h$ pointwise, then their intersection is always the same set, ensuring non-emptiness. □

Theorem 4.21. *Not every weakly associative triple (f, g, h) is strictly associative.*

Proof. Construct a counterexample by defining hyperfunctions where $(f \circ (g \circ h))(x)$ and $((f \circ g) \circ h)(x)$ differ but still overlap by at least one element. Such a construction shows that weak associativity does not enforce full equality, only a non-empty intersection. □

Theorem 4.22. *A collection of Weak Hyperfunctions forms a Weak Hyperstructure.*

Proof. Let $\mathcal{F}$ be a collection of hyperfunctions $f : S \to \mathcal{P}(S)$ such that for all $f, g, h \in \mathcal{F}$ and all $x \in S$, the weak associativity condition holds:

$$(f \circ (g \circ h))(x) \cap ((f \circ g) \circ h)(x) \neq \emptyset.$$

To prove that $\mathcal{F}$ forms a Weak Hyperstructure:

- The domain of the hyperoperation is $\mathcal{P}(\mathcal{F})$, the powerset of $\mathcal{F}$, which aligns with the definition of a Weak Hyperstructure.
- The operation $\circ$ satisfies weak associativity, as it ensures a non-empty intersection of results regardless of the order of composition.

Thus, $(\mathcal{P}(\mathcal{F}), \circ)$ is a Weak Hyperstructure. □

We can generalize hyperfunctions to n-Superhyperfunctions by replacing the codomain $\mathcal{P}(S)$ with an n-th powerset $\mathcal{P}_n(S)$. Similarly, we extend weak associativity to the n-th level.

Definition 4.23 (n-Superhyperfunction (Recall))**.** For $n \geq 1$, an *n-Superhyperfunction* is a function:

$$f : \mathcal{P}_r(S) \to \mathcal{P}_n(S),$$

where $0 \leq r \leq n$ and $\mathcal{P}_n(S)$ denotes the n-th powerset of S. Composition of n-Superhyperfunctions is defined by iterating the composition definition through the appropriate powerset levels.

Definition 4.24 (n-Weak Superhyperfunction)**.** A triple (f, g, h) of n-Superhyperfunctions is *n-weakly associative* if for all $X \in \mathcal{P}_r(S)$:

$$(f \circ (g \circ h))(X) \cap ((f \circ g) \circ h)(X) \neq \emptyset.$$

If every triple of n-Superhyperfunctions chosen from a collection satisfies this condition, we call them n-Weak Superhyperfunctions.

Theorem 4.25. *Every strictly associative triple of n-Superhyperfunctions is n-weakly associative.*

Proof. Strict associativity at the n-th level ensures equality of results, hence their intersection is non-empty. □

Theorem 4.26. *Not every n-weakly associative triple of n-Superhyperfunctions is strictly associative.*

Proof. Again, by constructing suitable examples where the sets differ but still share at least one element, we demonstrate that n-weak associativity does not imply strict equality. □

Theorem 4.27. *A collection of n-Weak Superhyperfunctions forms an n-Weak Superhyperstructure.*

Proof. Let $\mathcal{F}_n$ be a collection of n-Superhyperfunctions $f : \mathcal{P}_r(S) \to \mathcal{P}_n(S)$ for $0 \leq r \leq n$. Assume that for all $f, g, h \in \mathcal{F}_n$ and all $X \in \mathcal{P}_r(S)$, the n-weak associativity condition holds:

$$(f \circ (g \circ h))(X) \cap ((f \circ g) \circ h)(X) \neq \emptyset.$$

To prove that $\mathcal{F}_n$ forms an n-Weak Superhyperstructure:

- The domain of the hyperoperation is $\mathcal{P}_n(\mathcal{F}_n)$, the n-th powerset of $\mathcal{F}_n$, consistent with the definition of an n-Weak Superhyperstructure.
- The operation $\circ$ satisfies n-weak associativity, ensuring a non-empty intersection of results at the n-th level of composition.

Therefore, $(\mathcal{P}_n(\mathcal{F}_n), \circ)$ is an n-Weak Superhyperstructure. □

4.2 Discussions: Hypercontext

Other avenues for future exploration in this study is the examination of hypercontext. A context serves as a structured framework defining relationships, constraints, or patterns among elements within a specific domain or system (cf. [3, 73, 125, 140, 186, 304]). For instance, in conversations between individuals from different backgrounds, variations in context can lead to differences in perception or understanding. Context theory has also been studied from perspectives such as fuzzy theory [400, 432, 459, 530].

We aim to explore these contexts by extending them to hypercontexts and superhypercontexts. This includes investigating whether such extensions are implicitly implemented within existing systems, providing a broader perspective for analysis and application.

Definition 4.28 (Context)**.** Let U be a nonempty set, and let $R \subseteq U \times U$ be a relation on U. A context is defined as the pair $C = (U, R)$, where R represents contextual constraints or rules linking elements of U.

Example 4.29. Consider $U = \{\text{Person A}, \text{Person B}, \text{Person C}\}$, representing a group of individuals, and let $R = \{(\text{Person A}, \text{Person B}), (\text{Person B}, \text{Person C})\}$, representing a mentorship relationship. The context $C = (U, R)$ specifies who is mentoring whom in this setting.

Definition 4.30 (Hypercontext)**.** Given a context $C = (U, R)$, define a Hypercontext as:

$$\mathcal{HC} = (\mathcal{P}(U), R^*),$$

where $\mathcal{P}(U)$ is the power set of U, and $R^* \subseteq \mathcal{P}(U) \times \mathcal{P}(U)$ is a relation that extends R from individual elements to subsets of U. Instead of relating single elements, a hypercontext relates entire subsets, allowing us to capture more complex patterns or themes.

Theorem 4.31. *A Hypercontext generalizes a Context.*

Proof. Consider a context $C = (U, R)$. A Hypercontext is defined as $\mathcal{HC} = (\mathcal{P}(U), R^*)$. If we restrict $\mathcal{HC}$ to singletons $\{u\}$ for $u \in U$, the relation R^* on singletons can be chosen to recover the original relation R. Specifically, for $u, v \in U$, $(u, v) \in R$ if and only if $(\{u\}, \{v\}) \in R^*$. Thus, a Hypercontext, when restricted to singletons, behaves like a Context, showing that Hypercontexts generalize Contexts. □

Theorem 4.32. *A Hypercontext possesses the structure of a Hyperstructure.*

Proof. A Hyperstructure is built upon a powerset $\mathcal{P}(U)$ with operations defined on subsets. A Hypercontext $\mathcal{HC} = (\mathcal{P}(U), R^*)$ involves relations on $\mathcal{P}(U)$. To align it with a Hyperstructure, consider a hyperoperation $\circ$ defined on $\mathcal{P}(U)$ that interacts consistently with R^* (for example, defining $\circ$ to combine subsets in a way compatible with contextual constraints). This integration shows that a Hypercontext can be endowed with a suitable hyperoperation to form a Hyperstructure. Thus, a Hypercontext naturally fits into a Hyperstructure framework. □

Definition 4.33 (n-Superhypercontext)**.** For a given nonempty set U, let $\mathcal{P}_1(U) = \mathcal{P}(U)$ and define $\mathcal{P}_n(U) = \mathcal{P}(\mathcal{P}_{n-1}(U))$ as the n-th iterated power set of U. An n-Superhypercontext is defined as:

$$\mathcal{SHC}_n = (\mathcal{P}_n(U), R^{(n)}),$$

where $R^{(n)} \subseteq \mathcal{P}_n(U) \times \mathcal{P}_n(U)$ is a relation at the n-th power set level. This generalizes the concept of a hypercontext through multiple iterative layers, enabling analysis of contexts-of-contexts, and so on.

Theorem 4.34. *An n-Superhypercontext generalizes both a Hypercontext and a Context.*

Proof. An n-Superhypercontext $\mathcal{SHC}_n = (\mathcal{P}_n(U), R^{(n)})$ reduces to a Hypercontext $(\mathcal{P}(U), R^*)$ when $n = 1$. Thus, it generalizes a Hypercontext.

Since a Hypercontext generalizes a Context, and an n-Superhypercontext generalizes a Hypercontext, it follows by transitivity that an n-Superhypercontext also generalizes a Context. By choosing $n = 1$ and restricting to singletons, we recover the original Context structure. □

Theorem 4.35. *An n-Superhypercontext possesses the structure of an n-Superhyperstructure.*

Proof. By definition, an n-Superhypercontext $\mathcal{SHC}_n = (\mathcal{P}_n(U), R^{(n)})$ is defined at the n-th powerset level. An n-Superhyperstructure is also built on $\mathcal{P}_n(U)$ with operations defined at that level. Since $\mathcal{SHC}_n$ can be complemented by defining suitable n-th level hyperoperations that interact with $R^{(n)}$ to maintain structural properties, it inherits the n-Superhyperstructural form. In other words, just as a Hypercontext aligns with Hyperstructure at $n = 1$, an n-Superhypercontext aligns with an n-Superhyperstructure at the n-th level. □

Example 4.36. Consider U as a set of words, and R indicates that two words appear together frequently in scientific articles. A context $C = (U, R)$ can tell us if the word "cell" relates to the word "microscope."

A hypercontext $\mathcal{HC}$, on the other hand, works with subsets. For instance, it can describe whether the subset $\{\text{cell}, \text{microscope}, \text{tissue}\}$ is contextually related to $\{\text{protein}, \text{enzyme}\}$ in certain scientific fields. This could represent that both groups of words often appear together in the same kind of biological research articles.

An n-Superhypercontext $\mathcal{SHC}_n$ would then consider even more complex layers. For example, it might not only relate sets of words but also sets of sets of words, capturing patterns of usage across multiple scientific disciplines. At higher layers, you might discover that certain clusters of vocabulary sets are commonly found together in collections of research fields, showing a meta-level relationship among entire linguistic domains.

Example 4.37. Let U be a set of cities, and R says that two cities are connected by a direct flight. A context $C = (U, R)$ can tell us if city A is directly reachable from city B.

A hypercontext $\mathcal{HC}$ considers subsets of cities. It might describe whether the set of cities $\{A, B, C\}$ is related to another set $\{D, E\}$ through a pattern of flight connections. For example, $\{A, B, C\}$ might form a cluster of cities that are all connected to each other, and $\{D, E\}$ might form another cluster. If R^* relates these two clusters, it suggests there is a known travel pattern between these groups of cities (like a group of European cities often connected to a group of Asian cities).

An n-Superhypercontext $\mathcal{SHC}_n$ would then analyze relationships at multiple levels. At the second level, one might relate sets of city-clusters to other sets of city-clusters. Eventually, at higher n, one could analyze how different global flight networks (each represented as a set of city-clusters) interrelate, showing patterns in international travel and trade routes.

Example 4.38. Consider U as a set of academic courses, and R indicates that two courses are often taken by the same student in the same semester. A context $C = (U, R)$ can show whether Course 101 is related to Course 202 (i.e., often co-enrolled).

A hypercontext $\mathcal{HC}$ would look at subsets of courses. For example, the set {Course 101, Course 202} might be related to {Course 303, Course 404}, indicating that two groups of courses tend to form particular "curricular clusters."

An n-Superhypercontext $\mathcal{SHC}_n$ could then describe relationships among sets of these clusters, revealing patterns in entire academic programs or departments. At this higher level, we might find that certain departments (represented as sets of course subsets) often align with certain other departments' course groupings, helping university administrators understand inter-departmental academic flows.

4.3 Discussions: HyperVariable

A variable is a symbol representing an element from a set, widely used in mathematics and science [215, 252, 463, 511, 522]. HyperVariables are also well-established concepts [191]. We define the extended notion of superhypervariable and aim to explore its mathematical structure in future studies.

We anticipate further advancements in research on these concepts.

Definition 4.39 (Variable)**.** [215, 252] Let X be a non-empty set. A *variable* v is a symbol that can represent an element from X. Formally, $v \in X$. For example, if $X = \{x_1, x_2, \ldots, x_n\}$, then a variable v may take the value of any $x_i \in X$.

Example 4.40 (Variable)**.** Let $X = \{x, y, z\}$. A variable v could be $v = x$ at one time, and $v = z$ at another. Each realization of v picks out exactly one element of X.

Definition 4.41 (Hypervariable)**.** [191] Let X be a non-empty set and $\mathcal{P}(X)$ its powerset. A *hypervariable* V is a symbol that can represent a subset of X. Formally, $V \in \mathcal{P}(X)$. Unlike a variable which corresponds to a single element of X, a hypervariable represents a subset, allowing for more complex and flexible representations.

Example 4.42 (Hypervariable)**.** Let $X = \{1, 2, 3\}$. A hypervariable V could take values like:

$$V = \{1\}, \quad V = \{1, 2\}, \quad \text{or } V = \{2, 3\}.$$

Unlike a variable, which selects a single element, V selects subsets of various sizes.

Theorem 4.43. *A hypervariable possesses the structure of a Hyperstructure.*

Proof. A hypervariable takes values in $\mathcal{P}(X)$. Since a Hyperstructure is a mathematical framework built upon $\mathcal{P}(X)$ (often with hyperoperations defined on subsets), the domain of a hypervariable aligns naturally with the fundamental domain of Hyperstructures. By associating a hypervariable with the subsets of X, one can define hyperoperations on these subsets (for instance, unions, intersections, or more complex hyperoperations). This shows that the set of possible values of a hypervariable can be endowed with hyperoperations, making it compatible with a Hyperstructure framework. □

Theorem 4.44. *A hypervariable generalizes a variable.*

Proof. A variable v takes values in X. A hypervariable V takes values in $\mathcal{P}(X)$. Consider a hypervariable V that is always restricted to singleton sets, i.e., for all instances $V = \{x\}$ where $x \in X$. Under this restriction, each realization of V corresponds uniquely to an element of X, emulating a variable. Thus, a variable is a special case of a hypervariable confined to singleton sets. Hence, hypervariables strictly generalize variables. □

Definition 4.45 (n-Superhypervariable)**.** Let X be a non-empty set, and let $\mathcal{P}_n(X)$ denote the n-th powerset of X. An *n-superhypervariable* W_n is a symbol that can represent an element of $\mathcal{P}_n(X)$. Formally, $W_n \in \mathcal{P}_n(X)$. By iterating the powerset construction n-times, the n-superhypervariable can represent highly structured and hierarchical collections of subsets.

Example 4.46 (n-Superhypervariable)**.** For $n = 2$ and $X = \{a, b\}$, an element of $\mathcal{P}_2(X)$ might be:

$$W_2 = \{\emptyset, \{a\}\} \quad \text{or} \quad W_2 = \{\{a, b\}, \{b\}\}.$$

Here, W_2 picks elements from the second-level powerset, representing sets of subsets of X.

Theorem 4.47. *An n-superhypervariable generalizes a hypervariable (and consequently a variable).*

Proof. An n-superhypervariable W_n takes values in $\mathcal{P}_n(X)$. For $n = 1$, $\mathcal{P}_1(X) = \mathcal{P}(X)$, and thus an n-superhypervariable reduces to a hypervariable. Since we have already shown that a hypervariable generalizes a variable, it follows that an n-superhypervariable generalizes both a hypervariable and a variable. In other words, by setting $n = 1$ and restricting the n-superhypervariable to singleton sets at level $n = 1$, we recover the notion of a variable. Hence, n-superhypervariables are strictly more general constructs. □

Theorem 4.48. *An n-superhypervariable possesses the structure of an n-Superhyperstructure.*

Proof. An n-superhypervariable takes values in $\mathcal{P}_n(X)$. An n-Superhyperstructure is constructed on $\mathcal{P}_n(X)$ with suitable n-th level hyperoperations. Since $\mathcal{P}_n(X)$ is the domain of the n-superhypervariable, any n-superhypervariable can be associated with operations at the n-th power set level that define an n-Superhyperstructure. Therefore, an n-superhypervariable fits naturally into the n-Superhyperstructure framework, just as a hypervariable fits into a Hyperstructure. □

We now incorporate randomness into these concepts, inspired by previous discussions on Random Functions, Hyperfunctions, and their random counterparts. Random Variables have been extensively studied in probability theory and various other fields (cf. [200, 297, 325, 336, 428, 429]).

Definition 4.49 (Random Variable)**.** [355, 398, 399, 423, 436] Let $(\Omega, \mathcal{F}, P)$ be a probability space. A *Random Variable* is a function $v : \Omega \to X$ such that for each $\omega \in \Omega$, $v(\omega) \in X$. This recovers the classical definition of a random variable from probability theory.

Example 4.50 (Random Variable)**.** Let $X = \{a, b\}$ and $\Omega = [0, 1]$ with uniform measure. Define $v : \Omega \to X$ by:

$$v(\omega) = \begin{cases} a & \text{if } \omega < 1/2, \\ b & \text{if } \omega \geq 1/2. \end{cases}$$

This random variable chooses between a and b with equal probability.

Definition 4.51 (Random Hypervariable)**.** A *Random Hypervariable* is a function:

$$V : \Omega \to \mathcal{P}(X)$$

such that for each $\omega \in \Omega$, $V(\omega) \subseteq X$. This allows for random selection of subsets rather than single elements, generalizing a random variable.

Example 4.52 (Random Hypervariable)**.** Let $X = \{1, 2, 3\}$. Define $V : \Omega \to \mathcal{P}(X)$ by:

$$V(\omega) = \begin{cases} \{1\} & \text{if } \omega < 1/3, \\ \{1, 2\} & \text{if } 1/3 \leq \omega < 2/3, \\ \{2, 3\} & \text{if } \omega \geq 2/3. \end{cases}$$

Now the outcome is a random subset of X. If restricted to always produce singletons, it would be a random variable.

Theorem 4.53. *A Random Hypervariable generalizes a Random Variable.*

Proof. Restricting a Random Hypervariable $V : \Omega \to \mathcal{P}(X)$ to always yield singleton subsets reproduces the behavior of a Random Variable $v : \Omega \to X$. Thus, every Random Variable is a special case of a Random Hypervariable. □

Theorem 4.54. *A Random Hypervariable inherits a hyperstructural framework when suitable hyperoperations are defined on subsets.*

Proof. Since at each ω, a Hypervariable aligns with Hyperstructures, introducing randomness does not break this alignment. The random parameter ω simply selects which subsets are chosen, but the underlying set of possible values and hyperoperations remain the same. Hence, a Random Hypervariable can be viewed as a random selection of elements in a Hyperstructure. □

Definition 4.55 (Random n-Superhypervariable)**.** A *Random n-Superhypervariable* is a function:

$$W_n : \Omega \to \mathcal{P}_n(X)$$

where each realization $W_n(\omega) \in \mathcal{P}_n(X)$. This adds multiple hierarchical layers of complexity combined with randomness.

Example 4.56 (Random n-Superhypervariable)**.** For $n = 2$, let $X = \{a, b\}$. Consider:

$$W_2(\omega) = \begin{cases} \{\emptyset, \{a\}\} & \text{if } \omega < 1/2, \\ \{\{a, b\}, \{b\}\} & \text{if } \omega \geq 1/2. \end{cases}$$

This is a random selection of an element in $\mathcal{P}_2(X)$, showing both hierarchical complexity and randomness.

Theorem 4.57. *A Random n-Superhypervariable generalizes both Random Hypervariables and Random Variables.*

Proof. Set $n = 1$ to reduce a Random n-Superhypervariable $W_n : \Omega \to \mathcal{P}_n(X)$ to a Random Hypervariable. Since a Random Hypervariable generalizes a Random Variable, by transitivity, a Random n-Superhypervariable generalizes both Random Hypervariables and Random Variables. □

Theorem 4.58. *A Random n-Superhypervariable inherits an n-Superhyperstructural framework.*

Proof. Similarly, since an n-superhypervariable aligns with n-Superhyperstructures, introducing randomness yields a Random n-Superhypervariable. This random selection occurs within the hierarchical n-th powerset framework, preserving the n-Superhyperstructural alignment. □

Question 4.59. Can we define Hyperrandom Forest and Superhyperrandom Forest as extensions of Random Forest [53, 64, 74, 284]?

4.4 Discussions: n-th Powerset Convolution

We aim to explore the application of Hyperstructures and Powersets in Convolutional Networks in the future. Convolutional Networks are deep learning models designed for processing structured data like images, using convolution operations to extract features [249, 313, 314, 339, 559, 578].

We define the notion of a powerset convolution [550] for set functions and extend it to an n-th powerset convolution for functions defined on higher-order powersets. Furthermore, we prove that the n-th powerset convolution generalizes the powerset convolution and inherits the structural properties of the n-th powerset. Please note that this work is in the conceptual stage and represents a theoretical discussion.

Definition 4.60 (Powerset Convolution)**.** [550] Let N be a finite base set and $s : \mathcal{P}(N) \to \mathbb{R}$ be a set function, where $\mathcal{P}(N)$ is the powerset of N.

Given a filter $h : \mathcal{P}(N) \to \mathbb{R}$ (also a set function), the *powerset convolution* $h * s : \mathcal{P}(N) \to \mathbb{R}$ is defined by:

$$(h * s)(A) = \sum_{Q \subseteq N} h(Q)\, s(A \setminus Q), \quad \forall A \subseteq N,$$

where $A \setminus Q$ denotes the set difference.

This definition ensures shift-equivariance with respect to certain operators T_Q defined by $(T_Q s)(A) = s(A \setminus Q)$, making $h * s$ a natural convolution-like operation for set functions.

Theorem 4.61. *The powerset convolution is well-defined and linear.*

Proof. Linearity follows from the definition: each $(h * s)(A)$ is a finite linear combination of values of s with coefficients from h. Since N is finite, all sums are finite, ensuring well-definedness. □

Now we generalize the concept to an n-th powerset. Let $P_n(N)$ denote the n-th powerset of N, defined recursively by:

$$P_1(N) = \mathcal{P}(N), \quad P_{n+1}(N) = \mathcal{P}(P_n(N)).$$

A function $s : P_n(N) \to \mathbb{R}$ assigns real values to elements of the n-th powerset of N.

To define an n-th powerset convolution, we need a suitable generalization of the difference operation to n-th level sets. For $n = 1$, the difference is the standard set difference. For $n > 1$, we consider a recursive approach.

Definition 4.62 (n-th Level Difference)**.** For $n = 1$, the difference is set difference: $A \setminus Q$ for $A, Q \subseteq N$.

Assume we have defined a difference operation at level $n - 1$. For $X, Y \in P_n(N)$, define:

$$X \ominus Y = \{Z \in P_{n-1}(N) \mid Z = X' \ominus Y' \text{ for some } X' \in X, Y' \in Y\}$$

with the understanding that at level 1, $\ominus$ is just $\setminus$. If needed, choose a canonical representative (e.g., minimal pairs) to ensure a well-defined operation. This construction ensures a well-defined, associative-like structure at level n.

Given this n-th level difference, we define:

Definition 4.63 (n-th Powerset Convolution)**.** Let $s : P_n(N) \to \mathbb{R}$ and $h : P_n(N) \to \mathbb{R}$. The *$n$-th powerset convolution $h *_n s : P_n(N) \to \mathbb{R}$* is defined by:

$$(h *_n s)(X) = \sum_{Y \in P_n(N)} h(Y)\, s(X \ominus Y), \quad \forall X \in P_n(N),$$

where $\ominus$ is the n-th level difference defined above.

Theorem 4.64. *The n-th powerset convolution generalizes the powerset convolution.*

Proof. When $n = 1$, $P_1(N) = \mathcal{P}(N)$, and the operation $\ominus$ reduces to standard set difference. Thus,

$$(h *_1 s)(A) = \sum_{Q \subseteq N} h(Q) s(A \setminus Q),$$

which is exactly the powerset convolution. Hence, for $n = 1$ we recover the powerset convolution. Therefore, the n-th powerset convolution generalizes the powerset convolution. □

Theorem 4.65. *The n-th powerset convolution respects the n-th powerset structure.*

Proof. By construction, $h *_n s$ is defined on $P_n(N)$ and uses an n-th level difference operation $\ominus$ that is well-defined for elements of $P_n(N)$. Since $P_n(N)$ is constructed iteratively from N, and each step involves taking the powerset, the convolution $h *_n s$ aligns with the hierarchical structure of $P_n(N)$. In other words, $h *_n s$ is closed under operations defined at the n-th powerset level, reflecting the n-th powerset structure. □

4.5 Discussions: HyperMatrix

A matrix is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns, used in various applications [106,150,203,245,315,554,565]. Matrices have been applied in numerous research fields [61]. Hypermatrix, an extension of matrices [363], is also studied and further generalized into superhypermatrix frameworks. Relevant definitions and theorems are provided below. We anticipate further advancements in the study of these concepts in future research.

Definition 4.66 (Matrix)**.** [54, 523] Let K be a field (or a skewfield) and consider two finite indexing sets $I = \{1, 2, \ldots, m\}$ and $J = \{1, 2, \ldots, n\}$. A *matrix* over K is a function:

$$M : I \times J \to K.$$

In other words, a matrix is an $m \times n$ array of elements from K. Each pair (i, j) maps to a single element $M(i, j) \in K$. This is the classical concept of a matrix from linear algebra.

Example 4.67. Consider $I = \{1, 2\}$, $J = \{1, 2, 3\}$, and let $K = \mathbb{R}$. A matrix $M : I \times J \to \mathbb{R}$ could be:

$$M = \begin{pmatrix} 1 & -2 & 0 \\ \pi & \sqrt{2} & 3 \end{pmatrix}.$$

This matrix assigns to each (i, j) a single real number.

Definition 4.68 (Hypermatrix)**.** (cf. [363]) A *hypermatrix* generalizes the concept of a matrix by allowing the entries to be subsets of K rather than single elements. Formally, a hypermatrix $\mathcal{M}$ over K is a function:

$$\mathcal{M} : I \times J \to \mathcal{P}(K),$$

where $\mathcal{P}(K)$ is the powerset of K. For each pair (i, j), $\mathcal{M}(i, j)$ is a subset of K. This structure enables multi-valued or hyperalgebraic behavior at each entry.

Example 4.69. Using the same I, J, K as before, a hypermatrix could be:

$$\mathcal{M}(1, 1) = \{1, 2\}, \quad \mathcal{M}(1, 2) = \{-1\}, \quad \mathcal{M}(1, 3) = \{0, 4\}$$

$$\mathcal{M}(2, 1) = \{\pi\}, \quad \mathcal{M}(2, 2) = \{\sqrt{2}, \sqrt{3}\}, \quad \mathcal{M}(2, 3) = \{2, 3, 5\}.$$

This hypermatrix assigns subsets of $\mathbb{R}$ to each (i, j).

Theorem 4.70. *A hypermatrix generalizes a matrix.*

Proof. A matrix $M : I \times J \to K$ outputs a single element of K per entry. A hypermatrix $\mathcal{M} : I \times J \to \mathcal{P}(K)$ outputs a subset of K per entry. If we restrict each subset to be a singleton, we recover the single-valued nature of a matrix. Thus, every matrix is a special case of a hypermatrix with singleton subsets, proving that hypermatrices generalize matrices. □

Theorem 4.71. *A hypermatrix inherits the structure of a Hyperstructure.*

Proof. Consider a hypermatrix $\mathcal{M}$. Each entry $\mathcal{M}(i, j)$ is an element of $\mathcal{P}(K)$. If we define suitable hyperoperations on these subsets (e.g., hyperaddition of entries as union of subsets, hypermultiplication as setwise combinations), the set of all possible hypermatrix entries forms a hyperalgebraic structure. Thus, hypermatrices can be viewed as hyperfunctions with indices, naturally embedding into a Hyperstructure where operations are defined entrywise and yield sets as results. □

Definition 4.72 (n-Superhypermatrix)**.** For $n \geq 1$, the *n-Superhypermatrix* generalizes hypermatrices by iterating the powerset construction n-times. Let $\mathcal{P}_n(K)$ be the n-th powerset of K. An n-superhypermatrix $\mathcal{M}_n$ is defined as:

$$\mathcal{M}_n : I \times J \to \mathcal{P}_n(K),$$

where each entry $\mathcal{M}_n(i, j)$ is an element of the n-th powerset $\mathcal{P}_n(K)$. Thus, each entry can represent a nested, multi-layered collection of subsets, providing even more hierarchical complexity.

Example 4.73. For $n = 2$, an entry of an n-superhypermatrix might look like:

$$\mathcal{M}_2(i, j) = \{\{1, 2\}, \{3\}, \{\{4\}, \{5, 6\}\}\} \in \mathcal{P}_2(K).$$

This exhibits a two-level nesting of subsets.

Theorem 4.74. *An n-superhypermatrix generalizes both hypermatrices and matrices.*

Proof. For $n = 1$, $\mathcal{M}_1 : I \times J \to \mathcal{P}(K)$ is a hypermatrix. Since a hypermatrix generalizes a matrix, it follows by transitivity that an n-superhypermatrix, for $n \geq 1$, generalizes hypermatrices and hence also matrices. By choosing $n = 1$ and restricting each entry to singletons, we get a matrix; by allowing multiple subsets, we get hypermatrices. Increasing n adds more complexity. □

Theorem 4.75. *An n-superhypermatrix inherits the structure of an n-Superhyperstructure.*

Proof. An n-superhypermatrix has entries in $\mathcal{P}_n(K)$, the n-th powerset of K. Since n-Superhyperstructures operate on $\mathcal{P}_n(K)$ with suitable hyperoperations, defining appropriate entrywise n-th level hyperoperations allows embedding the n-superhypermatrix into an n-superhyperstructural framework. Each entry, being from $\mathcal{P}_n(K)$, can undergo n-th level hyperoperations to produce new sets at the n-th level. Thus, n-superhypermatrices fit into n-superhyperstructures. □

We previously discussed Random Functions, Hyperfunctions, and n-Superhyperfunctions, as well as Randomness, Hyperrandomness, and n-Superhyperrandomness. We can similarly define random versions of matrices, hypermatrices, and n-superhypermatrices by introducing probability spaces and randomness in their entries:

- A Random Matrix [51, 79, 512, 524, 531]: $M : \Omega \times I \times J \to K$ is a function that, for each $\omega \in \Omega$, yields a classical matrix.
- A Random Hypermatrix(cf. [27]): $\mathcal{M} : \Omega \times I \times J \to \mathcal{P}(K)$ introduces randomness and hyperalgebraic behavior.
- A Random n-Superhypermatrix: $\mathcal{M}_n : \Omega \times I \times J \to \mathcal{P}_n(K)$ adds multiple hierarchical layers.

We can further consider hyperrandomness and n-superhyperrandomness in these contexts if the statistical properties of entries change non-stationarily and in complex patterns.

Theorem 4.76. *A Random Hypermatrix generalizes a Random Matrix.*

Proof. A Random Matrix assigns to each (ω, i, j) a single element of K. A Random Hypermatrix assigns to each (ω, i, j) a subset of K. Restricting subsets to singletons reduces a Random Hypermatrix to a Random Matrix, showing that the former generalizes the latter. □

Theorem 4.77. *An n-Superhypermatrix, if made random and allowed n-superhyperrandom behaviors, generalizes all previously defined notions (Random Matrix, Random Hypermatrix) and can be embedded into an n-Superhyperstructure.*

Proof. By combining the ideas established:

- n-Superhypermatrices generalize hypermatrices and matrices.
- Randomness and hyperrandomness can be introduced, giving random n-superhypermatrices.
- n-superhyperrandomness can be modeled in n-superhyperstructures (as shown in previously proven theorems).

Therefore, a random n-superhypermatrix with n-superhyperrandom characteristics encompasses the entire hierarchy of complexity and uncertainty discussed so far. It can be integrated into an n-Superhyperstructure by defining appropriate n-th level hyperoperations and probability spaces. Thus, it generalizes all lower constructs. □

4.6 Discussions: Cognitive HyperMap

A Cognitive Map is a directed graph representing concepts (nodes) and their causal relationships (edges) with weighted influences [45, 412, 456, 551]. Derived concepts such as Fuzzy Cognitive Maps [36, 302, 319, 395, 396, 433], Neutrosophic Cognitive Maps [14, 279, 360, 388, 421], and Dynamic Cognitive Maps [94, 352] have also been studied. This subsection presents extended definitions of these concepts using the frameworks of Hypersructures and Superhyperstructures. Future research is anticipated to explore the applications of these extended models.

Definition 4.78 (Cognitive Map)**.** (cf. [45, 412, 456, 551]) A *cognitive map* is a directed graph $G = (V, E, w)$, where:

- $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of nodes, each representing a concept or variable.
- $E \subseteq V \times V$ is a set of directed edges, where $(v_i, v_j) \in E$ represents a causal or relational influence of v_i on v_j.
- $w : E \to \mathbb{R}$ is a weighting function that assigns a weight $w(e)$ to each edge $e = (v_i, v_j)$, quantifying the strength and direction of the influence. Positive weights indicate reinforcement, while negative weights indicate inhibition.

Definition 4.79 (The state of Cognitive Map)**.** The state of a cognitive map can be described by a vector:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^n,$$

where x_i represents the activation level or intensity of concept v_i.

Definition 4.80 (Update Rule of Cognitive Map)**.** The evolution of states over time can be modeled using an update function, typically of the form:

$$\mathbf{x}(t+1) = f(\mathbf{W} \cdot \mathbf{x}(t)),$$

where:

- $\mathbf{W}$ is the adjacency matrix of G, with entries $W_{ij} = w((v_i, v_j))$, representing the weights of the edges.
- $f : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function applied element-wise to ensure bounded states.

Remark 4.81 (Special Cases of Cognitive Maps)**.** Several related concepts can be derived from the foundational idea of Cognitive Maps:

- *Fuzzy Cognitive Maps (FCMs) [36, 302, 319, 395, 396, 433]:* These extend cognitive maps by allowing activation levels x_i to take values in the interval $[0, 1]$ and employing fuzzy logic to represent and analyze relationships.
- *Dynamic Cognitive Maps [94, 352]:* These maps incorporate time-dependent weights $w : E \times \mathbb{R} \to \mathbb{R}$, enabling the modeling of adaptive or evolving relationships over time, reflecting dynamic systems.
- *Neutrosophic Cognitive Maps [14, 279, 360, 388, 421]:* These generalize cognitive maps by introducing neutrosophic weights $w : E \to [T, I, F]$, where T, I, and F represent the degrees of truth, indeterminacy, and falsity, respectively. They are particularly effective for capturing and analyzing uncertainty and conflicting relationships in complex systems.

Definition 4.82 (Cognitive HyperMap)**.** Let $V = \{v_1, v_2, \ldots, v_n\}$ be a finite set of concepts and let $\mathcal{P}(V)$ denote the powerset of V. A *Cognitive HyperMap* is defined as:

$$\mathcal{H} = (V, \mathcal{E}, w),$$

where:

1. V is a finite set of concepts (vertices).

2. $\mathcal{E} \subseteq \mathcal{P}(V)\times\mathcal{P}(V)$ is a set of directed hyperedges. Each hyperedge $(A, B) \in \mathcal{E}$ maps a subset of concepts $A \subseteq V$ to another subset $B \subseteq V$, indicating that collectively, the concepts in A influence the concepts in B.
3. $w : \mathcal{E} \to \mathbb{R}$ assigns a real-valued weight to each hyperedge, quantifying the overall influence from the group A to the group B.

A Cognitive HyperMap generalizes a cognitive map by allowing hyperedges to represent multi-concept-to-multi-concept influences, rather than strictly pairwise influences.

Remark 4.83 (Special Cases of Cognitive HyperMaps)**.** Several related concepts can be derived from the foundational idea of Cognitive HyperMaps, extending the principles of Cognitive Maps into the framework of Hyperstructures:

- *Fuzzy Cognitive HyperMaps (FCHMs):* These extend Cognitive HyperMaps by allowing activation levels x_i to take values in the interval $[0, 1]$ and employing fuzzy logic to represent relationships. The hyperedges and hyperweights generalize the relationships among multiple concepts.
- *Dynamic Cognitive HyperMaps (DCHMs):* These incorporate time-dependent hyperweights $w : \mathcal{E}\times\mathbb{R} \to \mathbb{R}$, enabling the modeling of adaptive or evolving relationships over time within the Hyperstructure framework.
- *Neutrosophic Cognitive HyperMaps (NCHMs):* These generalize Cognitive HyperMaps by introducing neutrosophic hyperweights $w : \mathcal{E} \to [T, I, F]$, where T, I, and F represent the degrees of truth, indeterminacy, and falsity, respectively. This extension is particularly suited for analyzing uncertainty and conflicting relationships in complex systems.

Example 4.84. Consider $V = \{v_1, v_2, v_3\}$. In a cognitive map, one might have edges:

$$v_1 \to v_2, \quad v_2 \to v_3, \quad v_1 \to v_3.$$

In a cognitive hypermap, we could have a hyperedge:

$$\{v_1, v_2\} \to \{v_3\}$$

with a weight $w(\{v_1, v_2\}, \{v_3\}) = 0.8$, indicating that v_1 and v_2 collectively influence v_3.

Theorem 4.85. *A Cognitive HyperMap generalizes a Cognitive Map.*

Proof. A cognitive map has edges $v_i \to v_j$. A cognitive hypermap allows hyperedges $A \to B$ for subsets $A, B \subseteq V$. If we restrict each hyperedge to relate singletons, we recover the directed edges of a cognitive map. Thus, a cognitive map is a special case of a cognitive hypermap. □

Theorem 4.86. *A Cognitive HyperMap inherits the structure of a Hyperstructure.*

Proof. A cognitive hypermap involves hyperedges $A \to B$ with $A, B \subseteq V$. Since hyperstructures are defined on powersets with hyperoperations, we can interpret each hyperedge or influence as a hyperoperation on subsets of V. Thus, a cognitive hypermap's domain and co-domain are subsets from $\mathcal{P}(V)$, aligning naturally with the hyperstructural framework. □

Definition 4.87 (Cognitive n-SuperhyperMap)**.** Let V_0 be a finite set of base concepts. Define the n-th iterated power set of V_0 recursively as:

$$\mathcal{P}^0(V_0) = V_0, \quad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\big(\mathcal{P}^k(V_0)\big),$$

where $\mathcal{P}(A)$ denotes the power set of the set A.

A *Cognitive n-SuperhyperMap* is a triple $\mathcal{H}_n = (V, \mathcal{S}_n, w^{(n)})$, where:

1. $V \subseteq \mathcal{P}^n(V_0)$ is the set of *superconcepts* (sometimes called supervertices). Each superconcept is an element of the n-th iterated power set $\mathcal{P}^n(V_0)$. Thus, a superconcept can represent:
   - A single base concept $v \in V_0$,
   - A subset of V_0,
   - A more complex hierarchical subset structure up to n-levels (i.e., $v \in \mathcal{P}^n(V_0)$).
2. $\mathcal{S}_n \subseteq \mathcal{P}^n(V_0)$ is the set of n-th level *superedges*. Each superedge $S \in \mathcal{S}_n$ connects multiple superconcepts across potentially different levels of hierarchy. A superedge captures higher-dimensional relationships that may involve several groups of concepts simultaneously.
3. $w^{(n)} : \mathcal{S}_n \to \mathbb{R}$ is a weighting function that assigns a real-valued weight $w^{(n)}(S)$ to each n-th level superedge S. These weights quantify the strength and nature of the influences or interactions within the conceptual framework.

A Cognitive n-SuperhyperMap with Superedges generalizes the concept of Cognitive Maps and Cognitive HyperMaps by introducing hierarchical, multi-dimensional relationships. The use of superedges allows for more explicit modeling of complex interactions between subsets of concepts, enabling advanced analysis of multi-layered systems.

Example 4.88 (Illustration of a Cognitive n-SuperhyperMap for $n = 2$)**.** Let $n = 2$ and $V_0 = \{a, b\}$. The iterated power sets are as follows:

- The first-level power set is:

$$\mathcal{P}^1(V_0) = \mathcal{P}(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}.$$

- The second-level power set is:

$$\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0)) = \{\emptyset, \{\emptyset\}, \{\{a\}\}, \{\{b\}\}, \{\{a, b\}\}, \{\emptyset, \{a\}\}, \dots\}.$$

In a Cognitive n-SuperhyperMap with $n = 2$, consider the following example:

- A second-level superedge connects the group $\{\emptyset, \{a\}\}$ to another group $\{\{a, b\}, \{b\}\}$.
- The weight of this superedge is:

$$w^{(2)}(\{\emptyset, \{a\}\}, \{\{a, b\}, \{b\}\}) = 1.2.$$

This structure represents a hierarchical influence where:

- The group $\{\emptyset, \{a\}\}$ (including the absence of a and the singleton $\{a\}$) affects the group $\{\{a, b\}, \{b\}\}$ (including the joint set $\{a, b\}$ and singleton $\{b\}$).
- The weight 1.2 quantifies the strength and importance of this influence in the decision-making or conceptual framework.

Such an example illustrates how Cognitive n-SuperhyperMaps can capture multi-level, complex interdependencies among concepts, enabling advanced modeling of hierarchical systems.

Remark 4.89 (Specialized Variants of Cognitive n-SuperhyperMaps)**.** The foundational concept of Cognitive n-SuperhyperMaps can be extended into specialized frameworks, adapting the principles of Cognitive Maps and Cognitive HyperMaps to the n-Superhyperstructure context. Key variants include:

- *Fuzzy Cognitive n-SuperhyperMaps (FCnSHMs):* These variants extend Cognitive n-SuperhyperMaps by allowing activation levels x_i to range within the interval $[0, 1]$. Fuzzy logic governs the relationships, and n-th level superedges and weights generalize the interactions between superconcepts across multiple hierarchical layers.

- *Dynamic Cognitive n-SuperhyperMaps (DCnSHMs):* These models incorporate time-dependent n-th level weights $w^{(n)} : \mathcal{E}_n \times \mathbb{R} \to \mathbb{R}$, enabling the representation of dynamic, evolving relationships within the multi-level superhyperstructure. This approach captures temporal changes and adaptive behaviors in complex systems.
- *Neutrosophic Cognitive n-SuperhyperMaps (NCnSHMs):* These generalizations introduce neutrosophic weights $w^{(n)} : \mathcal{E}_n \to [T, I, F]$, where T, I, and F represent degrees of truth, indeterminacy, and falsity. This variant is particularly suited for modeling uncertainty, ambiguity, and contradictory hierarchical relationships in multi-layered systems.

These specialized frameworks highlight the adaptability and robustness of the Cognitive n-SuperhyperMap framework in addressing increasingly complex and uncertain conceptual structures.

Theorem 4.90. *A Cognitive n-SuperhyperMap generalizes a Cognitive HyperMap (and therefore also a Cognitive Map).*

Proof. For $n = 1$, a Cognitive n-SuperhyperMap reduces to a Cognitive HyperMap. Since a Cognitive HyperMap generalizes a Cognitive Map, it follows by transitivity that Cognitive n-SuperhyperMaps also generalize Cognitive Maps. Thus, by increasing n, we obtain higher-order structural complexity that encompasses the simpler forms as special cases. □

Theorem 4.91. *A Cognitive n-SuperhyperMap inherits the structure of an n-Superhyperstructure through the use of superedges.*

Proof. Cognitive n-SuperhyperMaps are constructed on $\mathcal{P}^n(V_0)$, which corresponds to the n-th iterated powerset of the base concept set V_0. In this framework:

- The superconcepts (nodes) are elements of $\mathcal{P}^n(V_0)$, representing hierarchical or multi-level subsets of V_0.
- The relationships between these superconcepts are encoded by n-th level superedges, which can connect subsets of $\mathcal{P}^n(V_0)$ at multiple levels.

An n-th level superedge is defined as a generalized relationship:

$$\mathcal{E}_n \subseteq \bigcup_{i,j=0}^{n} \mathcal{P}^i(V_0) \times \mathcal{P}^j(V_0),$$

allowing connections across different levels of hierarchy. These superedges, combined with the weighting function $w^{(n)}$, capture the multi-layered relationships and influences among superconcepts.

By aligning the structural definitions of Cognitive n-SuperhyperMaps with those of n-Superhyperstructures, we observe that:

1. The superconcepts correspond to the n-th level nodes in the n-Superhyperstructure.
2. The superedges define the interactions and dependencies within and across levels, analogous to the n-th level hyperoperations in n-Superhyperstructures.
3. The weighting function $w^{(n)}$ adds an additional layer of abstraction, quantifying the relationships represented by superedges.

Thus, the Cognitive n-SuperhyperMap adheres to the structural framework of an n-Superhyperstructure, with superedges and associated weights providing the necessary mathematical and conceptual connections. □

4.7 Discussions: Hyperfield

A field is a mathematical structure consisting of a set with two operations (addition and multiplication) satisfying associativity, commutativity, distributivity, and the existence of additive and multiplicative inverses [44, 148, 208, 294, 296, 328, 419]. A hyperfield is a generalization of a field where addition is a hyperoperation, producing a set of possible sums rather than a single value [271, 305, 347, 532].

Definition 4.92 (Field)**.** [282, 444, 462] A *field* is a set F equipped with two binary operations $+ : F \times F \to F$ (addition) and $\cdot : F \times F \to F$ (multiplication) such that:

1. $(F, +)$ is a commutative group with additive identity 0.
2. $(F \setminus \{0\}, \cdot)$ is a commutative group, ensuring every nonzero element is multiplicatively invertible.
3. Multiplication distributes over addition:

$$\forall a, b, c \in F, \quad a \cdot (b + c) = a \cdot b + a \cdot c \text{ and } (b + c) \cdot a = b \cdot a + c \cdot a.$$

A hyperfield generalizes a field by allowing the addition operation to be multivalued. While multiplication remains a standard binary operation, addition becomes a hyperoperation, yielding a subset of the field rather than a single element. This concept was originally introduced and studied by Krasner [271, 305, 347, 532].

Definition 4.93 (Hyperfield)**.** [271, 305, 347, 532] A *hyperfield* is a set X with two operations:

- A hyperaddition $\oplus : X \times X \to \mathcal{P}(X) \setminus \{\emptyset\}$, making $(X, \oplus)$ a commutative hypergroup.
- A usual (univalued) multiplication $\cdot : X \times X \to X$ such that:
  1. $(X \setminus \{0\}, \cdot)$ is a commutative group (so each nonzero element is multiplicatively invertible).
  2. Multiplication distributes over the hyperaddition in a strong sense:

$$a \cdot (b \oplus c) = a \cdot b \oplus a \cdot c, \quad (b \oplus c) \cdot a = b \cdot a \oplus c \cdot a.$$

The element 0 serves as the additive identity, and 1 is the multiplicative identity.

Theorem 4.94. *Any field is a hyperfield in which the hyperaddition reduces to a univalued addition.*

Proof. In a field F, the addition is a single-valued map $+ : F \times F \to F$. This trivially satisfies the hyperfield axioms if we interpret each sum $a+b$ as the singleton set $\{a+b\}$. The multiplicative structure and distributivity remain unchanged, making F a hyperfield with singleton-valued addition. □

Theorem 4.95. *Not every hyperfield is a field. Hyperfields strictly generalize fields by admitting multi-valued addition.*

Proof. Consider the Krasner hyperfield $K = \{0, 1\}$ where addition is defined by:

$$1 \oplus 1 = \{0, 1\}, \quad 1 \oplus 0 = \{1\}, \quad 0 \oplus 0 = \{0\}.$$

This addition is not univalued. Multiplication is standard with $0 \cdot x = 0$ and $1 \cdot x = x$. Such a structure cannot be a field because $1 \oplus 1$ is not a single element. Thus, hyperfields are strictly more general. □

To generalize further, we consider an n-th powerset construction analogous to n-Superhypergraphs and n-Superhyperstructures. Just as we defined n-Superhypergraphs by iterating power sets n-times, we now define n-Superhyperfields by lifting a hyperfield structure to higher-order powersets.

Definition 4.96 (n-Superhyperfield)**.** Let F be a hyperfield. Define $\mathcal{P}^0(F) = F$ and $\mathcal{P}^{k+1}(F) = \mathcal{P}(\mathcal{P}^k(F))$. An *$n$-Superhyperfield* is constructed as follows:

1. The underlying set of the n-Superhyperfield is $\mathcal{P}_n(F) = \mathcal{P}^n(F)$.
2. Addition at the n-th level, $\oplus_n : \mathcal{P}_n(F) \times \mathcal{P}_n(F) \to \mathcal{P}(\mathcal{P}_n(F)) \setminus \{\emptyset\}$, is defined by appropriately lifting the hyperaddition of F to the n-th power set. This creates a highly nested hyperoperation, where each addition step involves unions and expansions based on the previous power set level.
3. Multiplication $\cdot_n : \mathcal{P}_n(F) \times \mathcal{P}_n(F) \to \mathcal{P}_n(F)$ is defined similarly by lifting the multiplication of F, ensuring an invertible multiplicative structure at each nonzero level.
4. The distributivity conditions are extended to the n-th level, requiring that for any $A, B, C \in \mathcal{P}_n(F)$:

$$A \cdot_n (B \oplus_n C) \subseteq A \cdot_n B \oplus_n A \cdot_n C, \quad (B \oplus_n C) \cdot_n A \subseteq B \cdot_n A \oplus_n C \cdot_n A,$$

   and these conditions are strengthened as one ascends the n-th hierarchy to ensure a proper n-superhyperfield structure.

Intuitively, an n-Superhyperfield is a hyperfield at the n-th iterative power set level. At $n = 1$, we recover a hyperfield. At $n = 0$, restricting to singletons, we get a field. Thus, n-Superhyperfields generalize both hyperfields and fields.

Theorem 4.97. *An n-Superhyperfield generalizes a hyperfield (and thus also a field).*

Proof. For $n = 1$, the definition of an n-Superhyperfield reduces to that of a hyperfield. Since hyperfields generalize fields, it follows that n-Superhyperfields also generalize fields.

For higher n, each iteration of the powerset and hyperoperation lifts the structure to a higher complexity level. This nesting preserves generalization. Thus, n-Superhyperfields strictly encompass hyperfields and fields as special cases. □

Theorem 4.98. *The construction of an n-Superhyperfield from a given hyperfield F is well-defined and results in a structure that satisfies the n-Superhyperfield axioms.*

Proof. The construction is by induction on n:

- Base case ($n = 1$): $\mathcal{P}^1(F) = \mathcal{P}(F)$ with appropriately defined addition and multiplication is a hyperfield by definition.
- Inductive step: Assume $\mathcal{P}^n(F)$ forms an n-Superhyperfield. Consider $\mathcal{P}^{n+1}(F) = \mathcal{P}(\mathcal{P}^n(F))$. By applying the hyperoperations at the n-th level and extending them to the $(n + 1)$-th level sets, we ensure each operation remains closed, associative in the weak hyperfield sense, and that every nonzero element at the $(n + 1)$-th level is invertible under multiplication. Distributivity conditions can be verified by carefully lifting the conditions from the n-th level.

Thus, by induction, the n-Superhyperfield structure holds for all n. □

4.8 Discussions: Hypermodules

Modules generalize vector spaces by replacing the underlying field with a ring [8, 43, 62, 311, 457, 513]. Hypermodules further extend modules by allowing certain operations (notably addition) to be hyperoperations [378, 567], and superhypermodules push this construction to n-th iterated generalized powerset levels. In this section, we define these concepts rigorously and establish several results and theorems that illustrate their properties.

Definition 4.99 (Module)**.** (cf. [8, 43, 62, 311, 457]) Let R be a ring (with multiplicative identity 1) and let $(M, +)$ be an abelian group. Suppose there is an operation $\cdot : R \times M \to M$ called *scalar multiplication*, satisfying:

1. Distributivity over M: For all $r \in R$ and $x, y \in M$,

$$r \cdot (x + y) = r \cdot x + r \cdot y.$$

2. Distributivity over R: For all $r, s \in R$ and $x \in M$,

$$(r + s) \cdot x = r \cdot x + s \cdot x.$$

3. Associativity: For all $r, s \in R$ and $x \in M$,

$$(rs) \cdot x = r \cdot (s \cdot x).$$

4. Identity: For all $x \in M$,

$$1 \cdot x = x.$$

If these conditions hold, we call M a *left R-module*. A *right R-module* is defined analogously with the scalar multiplication on the right. If R is commutative, left and right modules coincide, and we often simply say "module."

Modules generalize vector spaces by replacing the field with a ring R. Unlike vector spaces, modules need not have a basis, and their structure can be more complex.

Definition 4.100 (Hypermodule)**.** (cf. [122, 378, 567, 575]) Let R be a ring and $(M, \oplus)$ be a commutative hypergroup (so $\oplus : M \times M \to \mathcal{P}(M) \setminus \{\emptyset\}$ is a hyperoperation making M a commutative hypergroup). Suppose we have a scalar multiplication $\cdot : R \times M \to M$ that is univalued but must interact with the hyperaddition $\oplus$ in a manner analogous to modules:

1. For all $r \in R$ and $x, y \in M$,
$$r \cdot (x \oplus y) \subseteq (r \cdot x) \oplus (r \cdot y).$$
2. For all $r, s \in R$ and $x \in M$,
$$(r + s) \cdot x \subseteq (r \cdot x) \oplus (s \cdot x).$$
3. For all $r, s \in R$ and $x \in M$,
$$(rs) \cdot x = r \cdot (s \cdot x) \quad \text{(associativity as in modules, still univalued for scalar multiplication),}$$
4. For all $x \in M$,
$$1 \cdot x = x.$$

If these conditions hold, we call M a *Hypermodule* over R.

In a Hypermodule, addition is multi-valued, but scalar multiplication remains a single-valued operation distributing over the hyperaddition in a "weak" sense. Every module is a hypermodule with singleton-valued addition.

Definition 4.101 (n-Superhypermodule)**.** Consider a Hypermodule M over R. Suppose $\mathcal{P}^n(M)$ (or a generalized construction $G_n(M)$ if we choose a generalized framework) is defined by iterating the powerset construction n-times. Define:

$$\oplus_n : \mathcal{P}_n(M) \times \mathcal{P}_n(M) \to \mathcal{P}(\mathcal{P}_n(M)) \setminus \{\emptyset\}$$

as an n-th level hyperaddition and similarly extend scalar multiplication:

$$\cdot_n : R \times \mathcal{P}_n(M) \to \mathcal{P}_n(M).$$

An *n-Superhypermodule* is a structure $(\mathcal{P}_n(M), \oplus_n, \cdot_n)$ where:

- $(\mathcal{P}_n(M), \oplus_n)$ is an n-superhyperstructure (i.e., a hypergroup at the n-th level).
- Scalar multiplication $\cdot_n$ is defined so that it "distributes" over $\oplus_n$ in a similar manner to a Hypermodule, but now at the n-th level.

Thus, an n-Superhypermodule generalizes a Hypermodule (and thus a module) by iteratively increasing complexity through n-th powersets or generalized n-th powersets.

Theorem 4.102. *Every module is a hypermodule (with singleton hyperaddition), and every hypermodule is a special case of an n-superhypermodule with $n = 1$.*

Proof. Module to Hypermodule: A module M has a single-valued addition $+$. Define $\oplus$ by $x \oplus y = \{x + y\}$. This trivially makes $(M, \oplus)$ a commutative hypergroup (since it's isomorphic to an abelian group). The scalar multiplication in the module already satisfies distributivity and associativity, so it still does for the hypermodule definition. Thus, every module is a hypermodule.

Hypermodule to n-Superhypermodule: For $n = 1$, define $(\mathcal{P}^1(M), \oplus_1)$ where $\oplus_1$ is just $\oplus$ lifted to subsets. If we restrict ourselves to singleton subsets and the induced operations, we recover the hypermodule. Therefore, every hypermodule can be embedded into an n-superhypermodule structure by considering $n = 1$. □

Theorem 4.103. *Consider a hypermodule M over R. If M possesses additional structural properties (e.g., free hypermodule with a "basis" in some generalized sense), these can be iterated to produce an n-superhypermodule inheriting analogous properties at the n-th level.*

Proof. Assume M is a hypermodule with a certain property P. Constructing the n-superhypermodule involves applying the powerset operation n-times. If property P is preserved under the formation of hyperoperations at the powerset level (for example, if P relates to closure properties, distributive axioms that scale with set operations, or structural decompositions that remain meaningful under iteration), then the induced n-superhypermodule also possesses P. The induction proceeds by verifying that each step from $\mathcal{P}^k(M)$ to $\mathcal{P}^{k+1}(M)$ preserves or extends P. □

Theorem 4.104. *If a hypermodule M over a ring R is finitely generated or free (in a certain generalized sense), then its n-superhypermodule construction $\mathcal{P}_n(M)$ inherits a notion of generability or freeness at the n-th level, albeit in a more complex form.*

Proof. Consider a hypermodule M generated by a finite set $\{m_1, \ldots, m_k\}$. Under the iteration for an n-superhypermodule, the generators become subsets at n-th level. The operation $\oplus_n$ allows any n-th level element to be expressed via scalar multiplication and hyperadditions of these generators, though now each "sum" may correspond to multiple sets. By carefully lifting the generability conditions through each powerset iteration, we maintain a notion of generation at the higher level. The proof is constructive, applying induction on n, and relies on verifying that every element in $\mathcal{P}_n(M)$ can be represented by a "hyper-basis" at that level. □

4.9 Discussions: Hyperlattices

In this subsection, we rigorously define lattices [108, 371, 460, 560], hyperlattices, and n-superhyperlattices. Lattices are algebraic structures where any two elements have a unique supremum (join) and infimum (meet), satisfying associativity, commutativity, and idempotency. We then state and prove several theorems illustrating their fundamental properties and relationships.

Definition 4.105 (Lattice)**.** [209, 371, 460, 560] A *lattice* is an algebraic structure $(L, \wedge, \vee)$ consisting of a non-empty set L equipped with two binary operations:

$$\wedge : L \times L \to L, \quad \vee : L \times L \to L,$$

called the *meet* and *join*, respectively, satisfying the following axioms for all $a, b, c \in L$:

1. *Commutativity:* $a \wedge b = b \wedge a$ and $a \vee b = b \vee a$.

2. *Associativity:* $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ and $(a \vee b) \vee c = a \vee (b \vee c)$.
3. *Absorption:* $a \wedge (a \vee b) = a$ and $a \vee (a \wedge b) = a$.

The following definitions, which will be used in the subsequent proofs of theorems, are introduced below.

Definition 4.106 (Ideal)**.** [67, 210] Let $L = (L, \vee, \wedge)$ be a lattice. A non-empty subset $I \subseteq L$ is called an *ideal* if:

- $a, b \in I$ implies $a \vee b \in I$ (closed under join),
- $a \in I$ and $b \leq a$ implies $b \in I$ (downward closed).

Definition 4.107 (Prime Ideal)**.** [67, 210] An ideal $P \subseteq L$ is a *prime ideal* if:

- $a \wedge b \in P$ implies $a \in P$ or $b \in P$.

Definition 4.108 (Maximal Ideal)**.** [67, 210] An ideal $M \subseteq L$ is a *maximal ideal* if:

- $M \neq L$,
- There is no ideal I such that $M \subsetneq I \subsetneq L$.

Definition 4.109 (Filter)**.** (cf. [67, 210]) A non-empty subset $F \subseteq L$ is called a *filter* if:

- $a, b \in F$ implies $a \wedge b \in F$ (closed under meet),
- $a \in F$ and $a \leq b$ implies $b \in F$ (upward closed).

A hyperlattice generalizes a lattice by replacing one of the operations (usually the join or the meet) with a hyperoperation [152, 299, 427, 547, 558].

Definition 4.110 (Hyperlattice)**.** (cf. [152, 547]) Let L be a non-empty set. A *hyperlattice* is a triple $(L, \wedge, \circ)$ where:

- $\wedge : L \times L \to L$ is a binary operation (single-valued), often analogous to a meet operation.
- $\circ : L \times L \to \mathcal{P}(L) \setminus \{\emptyset\}$ is a hyperoperation (multi-valued), often analogous to a join operation, such that $(L, \circ)$ is a commutative hypergroup and $(L, \wedge)$ is a commutative semigroup.

A hyperlattice must satisfy certain adapted forms of the lattice axioms. Typically, we impose:

1. *Idempotency:* For all $a \in L$, we have $a \in a \circ a$ and $a \wedge a = a$.
2. *Commutativity:* For all $a, b \in L$, $\circ$ is commutative in the hypergroup sense, and $\wedge$ is commutative as usual.
3. *Associativity Conditions:* While $\wedge$ remains associative, $\circ$ must satisfy the associative-like property for a hyperoperation (i.e., a weak associativity condition that ensures coherence of the structure).
4. *Absorption-like conditions:* Adapted from lattices, ensuring that $\wedge$ and $\circ$ interact in a manner resembling absorption. For instance, $a \in a \circ (a \wedge b)$ and $a \in a \wedge (a \circ b)$.
5. *Weak distributivity or s-distributivity (if required):* In a distributive hyperlattice, we might impose conditions like $a \wedge (b \circ c) \subseteq (a \wedge b) \circ (a \wedge c)$ or even equality if s-distributivity is demanded.

These conditions generalize those of a lattice to a setting where join is replaced by a hyperjoin.

To define an n-Superhyperlattice, we apply the n-th power set construction or a generalized n-th power set construction to a hyperlattice, lifting its elements to $\mathcal{P}^n(L)$ and adapting the operations accordingly.

Definition 4.111 (n-Superhyperlattice)**.** Let L be a hyperlattice. Define:

$$\mathcal{P}^0(L) = L, \quad \mathcal{P}^{k+1}(L) = \mathcal{P}(\mathcal{P}^k(L)), \text{ for } k \geq 0.$$

An *n-Superhyperlattice* is constructed as follows:

1. Its underlying set at level n is $\mathcal{P}^n(L)$.
2. The meet operation $\wedge_n : \mathcal{P}^n(L) \times \mathcal{P}^n(L) \to \mathcal{P}^n(L)$ is defined by lifting $\wedge$ in a suitable manner (e.g., elementwise or through a hypergroup construction at each level).
3. The join hyperoperation $\circ_n : \mathcal{P}^n(L) \times \mathcal{P}^n(L) \to \mathcal{P}(\mathcal{P}^n(L)) \setminus \{\emptyset\}$ is defined to generalize $\circ$ to the n-th level.

The axioms of hyperlattices must be adapted at each level n. The iterative construction ensures that at the n-th level, we have a structure where "elements" are now subsets of subsets ... of L up to n-fold iteration, and operations act in a correspondingly complex manner.

Theorem 4.112. *Every lattice $(L, \wedge, \vee)$ can be viewed as a hyperlattice by defining $a \circ b = \{a \vee b\}$. Consequently, every lattice is a special case of a hyperlattice.*

Proof. This is immediate by taking the join operation $\vee$ and turning it into a trivial hyperoperation with singleton outputs. The axioms of a hyperlattice reduce to those of a lattice. Hence, any lattice is a hyperlattice with no additional complexity in the join. □

Theorem 4.113. *Every hyperlattice can be embedded into an n-superhyperlattice for any $n \geq 1$.*

Proof. Given a hyperlattice $(L, \wedge, \circ)$, define $\mathcal{P}^n(L)$ and lift $\wedge$ and $\circ$ to $\wedge_n$ and $\circ_n$ at the n-th level. The resulting structure $(\mathcal{P}^n(L), \wedge_n, \circ_n)$ satisfies the axioms of an n-superhyperlattice. Taking $n = 1$ yields a structure isomorphic in some sense to the original hyperlattice (though more complicated), and larger n introduce hierarchical complexity. □

Theorem 4.114. *If $(L, \wedge, \circ)$ is a distributive hyperlattice (or s-distributive), then the induced operations at the n-th level preserve distributive properties, meaning $(\mathcal{P}^n(L), \wedge_n, \circ_n)$ remains distributive (or s-distributive).*

Proof. Distributivity conditions in hyperlattices typically involve inclusion relations like:

$$a \wedge (b \circ c) \subseteq (a \wedge b) \circ (a \wedge c).$$

At the n-th level, a, b, c become sets of sets, and we must apply these distributive laws elementwise, using induction. The base case $n = 1$ is given by the definition of hyperlattice distributivity. Assuming it holds at level n, we show that lifting from $\mathcal{P}^n(L)$ to $\mathcal{P}^{n+1}(L)$ preserves the distributive pattern because each operation at level $n + 1$ is defined in terms of unions and products of sets at level n. Thus, the distributive pattern is inductively maintained. □

Theorem 4.115. *If a hyperlattice $(L, \wedge, \circ)$ admits a theory of ideals, prime ideals, and maximal ideals, then its induced n-superhyperlattice $(\mathcal{P}^n(L), \wedge_n, \circ_n)$ also admits analogous concepts (e.g., prime and maximal "hyperideals") at the n-th level.*

Proof. The notions of ideals (or filters) in a hyperlattice rely on closure properties with respect to $\wedge$ and conditions relating to $\circ$. At the n-th level, these notions translate to subsets of $\mathcal{P}^n(L)$ stable under the operations $\wedge_n, \circ_n$. The definition of hyperideals at level n essentially lifts the conditions from level $n - 1$, ensuring prime and maximal ideals (if they exist) also ascend to higher levels. Detailed verification requires checking that upward closures and downward closures under $\wedge_n, \circ_n$ remain well-defined and non-trivial, which follows from the inductive construction of these operations. □

Theorem 4.116. *Hypermodules possess the structure of a Hyperstructure. Similarly, n-Superhypermodules possess the structure of a Superhyperstructure.*

Proof. We will prove the two claims sequentially:

1. *Hypermodules as Hyperstructures:* A Hypermodule $(M, \oplus, \cdot)$ consists of a commutative hypergroup $(M, \oplus)$ with scalar multiplication $\cdot$. By definition:

$$\oplus : M \times M \to \mathcal{P}(M) \setminus \{\emptyset\},$$

making $(M, \oplus)$ a hyperstructure. The scalar multiplication $\cdot$ interacts with $\oplus$ in a distributive manner:

$$r \cdot (x \oplus y) \subseteq (r \cdot x) \oplus (r \cdot y).$$

This satisfies the key properties of a Hyperstructure, with addition being multi-valued and scalar multiplication compatible with this multi-valued addition. Thus, a Hypermodule is a Hyperstructure.

2. *n-Superhypermodules as Superhyperstructures:* Consider an n-Superhypermodule $(\mathcal{P}_n(M), \oplus_n, \cdot_n)$, where $\mathcal{P}_n(M)$ is the n-th iterated powerset of M, and $\oplus_n$ is the n-th level hyperaddition:

$$\oplus_n : \mathcal{P}_n(M) \times \mathcal{P}_n(M) \to \mathcal{P}(\mathcal{P}_n(M)) \setminus \{\emptyset\}.$$

The scalar multiplication $\cdot_n : R \times \mathcal{P}_n(M) \to \mathcal{P}_n(M)$ distributes over $\oplus_n$ in a manner analogous to Hypermodules:

$$r \cdot (A \oplus_n B) \subseteq (r \cdot A) \oplus_n (r \cdot B), \quad \forall A, B \in \mathcal{P}_n(M).$$

Since $(\mathcal{P}_n(M), \oplus_n)$ forms an n-Superhyperstructure by definition, the entire structure $(\mathcal{P}_n(M), \oplus_n, \cdot_n)$ inherits the hierarchical complexity of Superhyperstructures. Thus, n-Superhypermodules are Superhyperstructures.

In both cases, the respective module extensions (Hypermodules and n-Superhypermodules) satisfy the structural requirements for Hyperstructures and Superhyperstructures, respectively. Therefore, the theorem holds. □

The concept of a Semilattice is well-known as a related notion to Lattices [76, 96, 416, 430]. A semilattice is a mathematical structure with a commutative, associative, idempotent binary operation over a non-empty set. This Semilattice is also extended to hyperconcepts and superhyperconcepts. The relevant definitions and theorems are provided below.

Definition 4.117 (Semilattice)**.** (cf. [76,96,416,430]) A *semilattice* is a pair $(S, *)$ where S is a non-empty set and $* : S \times S \to S$ is a binary operation satisfying:

1. *Commutativity:* For all $a, b \in S$, $a * b = b * a$.
2. *Associativity:* For all $a, b, c \in S$, $(a * b) * c = a * (b * c)$.
3. *Idempotency:* For all $a \in S$, $a * a = a$.

These axioms ensure that $(S, *)$ can be seen as a commutative, idempotent semigroup. Common examples of semilattices include sets of subsets of a given set under union or intersection.

Definition 4.118 (Semi-Hyperlattice)**.** [28, 29] A *semi-hyperlattice* is a pair $(S, \diamond)$ where S is a non-empty set and $\diamond : S \times S \to \mathcal{P}(S) \setminus \{\emptyset\}$ is a hyperoperation satisfying:

1. *Commutativity:* For all $a, b \in S$, $\diamond$ is commutative, i.e., $a \diamond b = b \diamond a$.

2. *Associativity (in the hyper sense):* For all $a, b, c \in S$,

$$\bigcup_{x \in a \diamond (b \diamond c)} x \cap \bigcup_{y \in (a \diamond b) \diamond c} y \neq \emptyset.$$

This is a weak associativity condition ensuring that no matter how we parenthesize the operation, the resulting sets have at least one common element.

3. *Idempotency:* For all $a \in S$, $a \in a \diamond a$.

Such a structure can be interpreted as a "join-only" or "meet-only" hyperstructure that generalizes a semilattice. Each binary combination of elements yields a set of possible "results," maintaining commutativity, a form of associativity, and idempotency.

Theorem 4.119. *Every semilattice $(S, *)$ is a special case of a semi-hyperlattice by defining $a \diamond b = \{a * b\}$. Thus, semilattices are embedded into the class of semi-hyperlattices.*

Proof. In a semilattice, the operation $*$ is single-valued. Define a hyperoperation $\diamond$ by $a \diamond b = \{a * b\}$. This trivially makes $(S, \diamond)$ a semi-hyperlattice since the hyperoperation now always returns a singleton set, preserving commutativity, associativity (in a trivial sense), and idempotency. Hence, every semilattice is a degenerate semi-hyperlattice. □

Definition 4.120 (Semi-n-Superhyperlattice)**.** Let $(S, \diamond)$ be a semi-hyperlattice. Define:

$$\mathcal{P}^0(S) = S, \quad \mathcal{P}^{k+1}(S) = \mathcal{P}(\mathcal{P}^k(S)), \text{ for } k \geq 0.$$

An *semi-n-superhyperlattice* is constructed as:

- Its underlying set at level n is $\mathcal{P}^n(S)$.
- The operation $\diamond_n : \mathcal{P}^n(S) \times \mathcal{P}^n(S) \to \mathcal{P}(\mathcal{P}^n(S)) \setminus \{\emptyset\}$ is defined by lifting $\diamond$ to the n-th level. This involves defining, for example:

$$A \diamond_n B = \bigcup_{a \in A, b \in B} (a \diamond b)$$

  or a more elaborate definition depending on the chosen approach. The key is that $\diamond_n$ must be commutative, weakly associative, and idempotent at the n-th level.

The axioms of a semi-hyperlattice must be suitably adapted so that the n-th level construction $(\mathcal{P}^n(S), \diamond_n)$ remains a semi-hyperlattice at the n-th order.

Theorem 4.121. *Every semi-hyperlattice can be embedded into a semi-n-superhyperlattice for any $n \geq 1$.*

Proof. Starting from a semi-hyperlattice $(S, \diamond)$, construct $\mathcal{P}^n(S)$ and define $\diamond_n$ as:

$$A \diamond_n B = \bigcup_{a \in A, b \in B} (a \diamond b).$$

We must verify the axioms at the n-th level:

- *Commutativity:* This follows from the commutativity of $\diamond$, as any pairwise combination is symmetric.
- *Weak associativity:* If $\diamond$ satisfies weak associativity at the base level, then at the n-th level, we must check that

$$\bigcup_{X \in A \diamond_n (B \diamond_n C)} X \cap \bigcup_{Y \in (A \diamond_n B) \diamond_n C} Y \neq \emptyset.$$

  This condition can be shown by induction on n. The base case $n = 1$ is the definition of the semi-hyperlattice. Assuming it holds for $\mathcal{P}^n(S)$, lifting to $\mathcal{P}^{n+1}(S)$ involves unions and set combinations that preserve non-empty intersection due to the structure of the hyperoperation.

- *Idempotency:* For any $A \in \mathcal{P}^n(S)$, since $\diamond$ is idempotent at level 0, we have for all $a \in A$, $a \in a \diamond a$. At level n, consider $A \diamond_n A$. By definition, for each $a, a' \in A$, $a \diamond a'$ is non-empty and if $a' = a$, $a \in a \diamond a$. This ensures $A \subseteq A \diamond_n A$ because for each $a \in A$, $a \in a \diamond a \subseteq A \diamond_n A$.

Thus, $(\mathcal{P}^n(S), \diamond_n)$ satisfies the axioms of a semi-hyperlattice at the n-th level. □

Theorem 4.122. *If a semi-hyperlattice $(S, \diamond)$ has additional distributive-like properties (e.g., a form of distributivity involving subsets), then these properties lift to the n-th level in a semi-n-superhyperlattice.*

Proof. The proof is analogous to the distributivity results in hyperlattices. Since we only have one operation $\diamond$, distributivity often refers to conditions involving other set-theoretic constructions or additional operations (like a meet operation if we consider expansions). If such conditions hold at the base level, they can be verified by induction to hold at higher levels because the n-th level operation $\diamond_n$ is defined in terms of the base operation $\diamond$. Each distributive condition translates into set inclusions or equalities that remain valid under unions and iterative power set constructions. □

Theorem 4.123. *If $(S, \diamond)$ is a semi-hyperlattice with a well-defined notion of "ideals" and "prime ideals" (or analogous concepts), then these notions also ascend to the n-th level $(\mathcal{P}^n(S), \diamond_n)$, yielding a theory of prime "hyperideals" at higher levels.*

Proof. The notion of an ideal in a semi-hyperlattice would involve closure properties under $\diamond$ and certain downward closure properties. When we move to the n-th level, an "ideal" becomes a subset of $\mathcal{P}^n(S)$ stable under $\diamond_n$. Prime-like conditions, i.e., conditions that replicate the behavior of prime ideals (where certain decompositions imply membership conditions), also lift naturally because they rely on the underlying structure of $\diamond$. The induction from level n to $n + 1$ ensures that prime conditions remain meaningful, as the operations and their axioms are consistently applied at each level. □

4.10 Discussions: Boolean Hyperalgebra

A Boolean algebra is a mathematical structure with operations (and, or, not) satisfying commutativity, associativity, distributivity, identity, and complement laws [164,221,222,365,418,477,479]. A Boolean hyperalgebra generalizes Boolean algebra by replacing binary operations with hyperoperations, allowing multi-valued results while retaining complement properties [239, 449, 507]. Boolean SuperHyperAlgebra extends Boolean hyperalgebra by incorporating m-ary operations and n-th powersets, enabling hierarchical, multi-valued logical frameworks for complex systems. Boolean SuperHyperAlgebra can also be understood as a concept applying the ideas of SuperHyperAlgebra, as explored in [499, 500], to the realm of Boolean algebra. Definitions and related theorems are provided below.

Definition 4.124 (Boolean Algebra)**.** [221,222,477] A *Boolean algebra* is a structure $(B, \wedge, \vee, ', 0, 1)$ where B is a non-empty set, and $\wedge$, $\vee$ are binary operations, and $'$ is a unary operation (complement), and 0 and 1 are distinguished elements of B, such that for all $x, y, z \in B$:

1. $(B, \wedge, 1)$ is a commutative, associative, idempotent structure with $x \wedge 1 = x$ and $x \wedge x = x$.

2. $(B, \vee, 0)$ is a commutative, associative, idempotent structure with $x \vee 0 = x$ and $x \vee x = x$.

3. Distributivity: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ and $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.

4. Complementation: For each $x \in B$, there is an $x' \in B$ such that $x \wedge x' = 0$ and $x \vee x' = 1$.

These axioms ensure that the algebraic structure mimics classical Boolean logic.

Definition 4.125 (Boolean Hyperalgebra)**.** [239,449,507] A *Boolean hyperalgebra* is a structure $(H, \circ_1, \circ_2, ', 0, 1)$ where:

1. H is a non-empty set.

2. $\circ_1, \circ_2 : H \times H \to P(H) \setminus \{\emptyset\}$ are hyperoperations (often generalizing $\wedge$ and $\vee$ from Boolean algebra) that are commutative, weakly associative, and idempotent.

3. There is a unary operation $'$: $H \to H$ providing complements such that for each $x \in H$:

$$\exists x' \in H \text{ with } x \circ_1 x' = \{0\} \text{ and } x \circ_2 x' = \{1\}$$

(interpreting these hyperoperations in a way that mimics the Boolean complement property).

4. 0 and 1 are special elements analogous to the least and greatest elements from Boolean algebra.

5. A form of distributivity or s-distributivity (depending on the chosen variant) ensures that the Boolean-like structure is preserved at the hyper level.

The axioms are chosen so that if we replace each hyperoperation by a single-valued operation (choosing singletons), we recover a Boolean algebra. Thus, a Boolean hyperalgebra is a hyperstructural generalization of a Boolean algebra.

Definition 4.126 (Boolean (m,n)-SuperHyperalgebra)**.** A *Boolean* (m,n)*-SuperHyperalgebra* is an algebraic structure:

$$\mathcal{A} = (H, \{\circ_i^{(m,n)}\}_{i \in I}, ', 0, 1)$$

where:

1. H is a non-empty set.

2. Each $\circ_i^{(m,n)}$ is an m-ary SuperHyperOperation:

$$\circ_i^{(m,n)} : H^m \to P_*^n(H) \text{ or } P^n(H),$$

depending on whether we are dealing with classical-type or Neutrosophic-type (allowing empty sets) superhyperoperations. The collection of indices I may represent a finite or infinite family of such operations.

3. At least one of these operations plays the role analogous to $\vee$ or $\wedge$ in the Boolean algebra setting, and must satisfy commutativity, weak associativity (or strong associativity if we specify), and idempotency.

4. A complement operation $'$: $H \to H$ exists such that for each $x \in H$:

x' is a complement satisfying Boolean-like conditions: $x \circ_i^{(m,n)} x'$ includes 0 and 1 appropriately.

The exact form of complement conditions may depend on how the Boolean hyperalgebra axioms are generalized to m-ary and n-th powerset contexts.

5. 0 and 1 remain distinguished elements (or sets) acting as the neutral and absorbing elements analogous to those in Boolean algebras.

6. A form of distributivity or generalized distributivity (depending on the complexity of $\circ_i^{(m,n)}$) is imposed to ensure the structure mimics Boolean logic at a higher complexity level.

This structure significantly generalizes both Boolean algebras and Boolean hyperalgebras by allowing multiple inputs (m-ary [35, 52, 121]) and hierarchical complexity (n-th powerset), thus yielding a complex system that can encode intricate logical relationships and uncertainties.

Theorem 4.127. *Every Boolean algebra* $(B, \wedge, \vee, ', 0, 1)$ *is a special case of a Boolean hyperalgebra where each hyperoperation outputs singletons. Further, each Boolean hyperalgebra is a special case of a Boolean* (m,n)*-SuperHyperalgebra*[1]*.*

[1]For $m = 2$, $n = 1$, and restricting hyperoperations to singletons.

Proof. If in a Boolean hyperalgebra we choose each hyperoperation so that it always returns a singleton set, we reduce it to a single-valued operation scenario. With appropriately chosen operations matching $\wedge$ and $\vee$, and a complement operation $'$, and given elements 0, 1, the structure collapses to a Boolean algebra.

Similarly, a Boolean hyperalgebra is obtained from a Boolean (m, n)-SuperHyperalgebra by choosing $m = 2$, $n = 1$, and restricting the hyperoperations to behave as binary hyperoperations at the base level with single-step powerset.

Thus the inclusion of these structures is evident. □

Theorem 4.128. *In a Boolean* (m, n)*-SuperHyperalgebra, for every element* $x \in H$*, there exists* $x' \in H$ *(the complement of* x*) such that* x *combined with* x' *through the relevant superhyperoperations yields subsets containing* 0 *and* 1*, mimicking the Boolean complement property.*

Proof. By definition, a Boolean (m, n)-SuperHyperalgebra is designed to generalize Boolean concepts. Therefore, it must retain the ability to "negate" or "complement" each element. The axioms ensure there is a unary operation $'$ that assigns a complement to each element. By construction, the complement operation in a Boolean (m, n)-SuperHyperalgebra ensures:

$$x \circ_i^{(m,n)} (x', \ldots, x') \supseteq \{0\}, \quad x \circ_j^{(m,n)} (x', \ldots, x') \supseteq \{1\}$$

for suitably chosen $i, j \in I$ indexing the operations. Hence complements exist and behave analogously to Boolean algebra complements. □

Theorem 4.129. *If a Boolean hyperalgebra satisfies a certain distributive or s-distributive property, then its corresponding Boolean* (m, n)*-SuperHyperalgebra can also be endowed with a similar distributive property at each level of iteration, provided the hyperoperations and their* m*-ary extensions are defined consistently.*

Proof. The distributivity or s-distributivity conditions in Boolean hyperalgebras state that certain set inclusions or equalities hold among the results of hyperoperations. When extending to Boolean (m, n)-SuperHyperalgebra, we define $\circ_i^{(m,n)}$ at n-th powerset levels. Since these constructions arise by applying power set operations and unions over the base hyperoperations, any set-theoretic property like distributivity can be verified via induction on n. The complexity of m-ary operations does not obstruct distributivity as it typically generalizes from binary to m-ary by imposing analogous conditions on all tuples of inputs. Thus, distributivity properties can be preserved. □

4.11 Discussions: Generalized n-th Powerset

In [180], the concept of a Generalized n-th Powerset was introduced as an extension of the traditional n-th Powerset. Building upon this, we anticipate that applying this framework to the concepts proposed in this paper will contribute to further advancements in mathematical structures and their practical applications. For readers interested in a detailed understanding of Fuzzy Sets or Neutrosophic Sets, we recommend referring to foundational lecture notes or introductory materials (e.g., [39, 190, 484]).

Definition 4.130 (Generalized n-th Powerset)**.** [180] Let H be a set or a mathematical structure, and let $P(H)$ denote the classical powerset of H. Define the n-th generalized powerset of H, denoted $G_n(H)$, recursively as:

$$G_1(H) = G(H),$$

$$G_{n+1}(H) = G\big(G_n(H)\big) \quad \text{for } n \geq 1,$$

where $G(H)$ is a generalized powerset operator that incorporates additional constraints, properties, or structures. Examples of $G(H)$ include:

- *Labeled subsets:* $G(H) = \{(A, \ell_A) \mid A \subseteq H, \ell_A \in L\}$, where L is a set of labels.
- *Weighted subsets [562]:* $G(H) = \{(A, w_A) \mid A \subseteq H, w_A \in \mathbb{R}\}$, where weights w_A are assigned to subsets.

- *Soft subsets [359]:* Let U be a universe and E a set of parameters. A soft subset over U is a pair (F, A), where $A \subseteq E$ and $F : A \to P(U)$. For each $e \in A$, $F(e) \subseteq U$ represents the set of elements satisfying parameter e.
- *Graph subsets:* $G(H) = \{(G, V_G, E_G) \mid V_G \subseteq V(H), E_G \subseteq E(H)\}$, where $G = (V_G, E_G)$ is a subgraph of H.
- *Structured subsets:* Subsets with internal structures, such as orderings, multisets, or graph-like properties.
- *Filtered subsets:* Subsets satisfying a predicate $P(A)$, such that $G(H) = \{A \subseteq H \mid P(A)\}$.
- *Fuzzy subsets [568]:* $G(H) = \{(A, \mu_A) \mid A \subseteq H, \mu_A : A \to [0, 1]\}$, where μ_A defines the degree of membership for each element in A.
- *Rough subsets [405]:* Defined in terms of lower and upper approximations, $G(H) = \{(A, \underline{A}, \overline{A}) \mid A \subseteq H\}$, where:

$$\underline{A} = \{x \in H \mid P(x) \text{ is definitely true}\}, \quad \overline{A} = \{x \in H \mid P(x) \text{ is possibly true}\}.$$

- *Neutrosophic subsets [481]:* $G(H) = \{(A, T_A, I_A, F_A) \mid A \subseteq H, T_A, I_A, F_A : A \to [0, 1]\}$, where:

$$0 \le T_A(x) + I_A(x) + F_A(x) \le 3 \quad \text{for all } x \in A,$$

  and $T_A(x)$, $I_A(x)$, and $F_A(x)$ represent the degrees of truth, indeterminacy, and falsity, respectively.
- *Plithogenic subsets [486, 503]:* $G(H) = \{(A, v, Pv, pdf, pCF) \mid A \subseteq H\}$, where:
  - v is an attribute.
  - Pv is the range of possible values for v.
  - $pdf : A \times Pv \to [0, 1]^s$ is the *Degree of Appurtenance Function (DAF)*.
  - $pCF : Pv \times Pv \to [0, 1]^t$ is the *Degree of Contradiction Function (DCF)* satisfying:

$$pCF(a, a) = 0, \quad pCF(a, b) = pCF(b, a) \quad \text{for all } a, b \in Pv.$$

Definition 4.131 (Generalized Non-Empty n-th Powerset)**.** [180] Define the n-th generalized non-empty powerset of H, denoted $G_n^*(H)$, recursively as:

$$G_1^*(H) = G^*(H),$$

$$G_{n+1}^*(H) = G^*\big(G_n^*(H)\big),$$

where $G^*(H)$ is the non-empty subset operator under the generalized powerset $G(H)$, satisfying $G^*(H) \subseteq G(H) \setminus \{\emptyset\}$.

Theorem 4.132. *A Generalized n-th Powerset generalizes a n-th Powerset.*

Proof. It is evident from the definition. □

Theorem 4.133. *A Generalized n-th Powerset generalizes a classic Powerset.*

Proof. It is evident from the definition. □

Theorem 4.134. *[180] The Generalized n-th Powerset can represent the structure of supervertices and superedges in an n-SuperHyperGraph.*

Proof. It is evident from the definition. Refer to [180] as needed. □

In this paper, we discussed classical hyperstructures, n-superhyperstructures, and the concepts of generalized n-th powersets. To further extend these notions, we now introduce the concepts of Generalized Hyperstructures and Generalized n-Superhyperstructures. These frameworks provide an even broader stage for incorporating additional attributes, logic, uncertainty, or complexity into the hyperstructural environment.

Definition 4.135 (Generalized Hyperstructure)**.** Let H be a set or a mathematical structure, and let $G(H)$ be a Generalized Powerset of H, as defined by a chosen operator G that assigns additional structure, such as fuzziness, neutrosophic values, weights, parameters, or other internal complexities.

A *Generalized Hyperstructure* is a pair:

$$\mathcal{G} = (G(H), \diamond),$$

where:

- $G(H)$ is a generalized collection of subsets or enriched subsets of H (e.g., fuzzy subsets, neutrosophic subsets, weighted subsets, etc.).
- $\diamond : G(H) \times G(H) \to \mathcal{P}(G(H)) \setminus \{\emptyset\}$ is a hyperoperation that takes two elements from $G(H)$ and returns a non-empty subset of $G(H)$.

  The conditions for associativity (strict or weak), commutativity, identity, and invertibility can be adapted depending on the intended generalization. The key point is that both the underlying domain and the hyperoperation are "generalized," potentially carrying additional attributes or logic.

Definition 4.136 (Generalized n-Superhyperstructure)**.** Let G be a generalized powerset operator, and define:

$$G_1(H) = G(H), \quad G_{n+1}(H) = G(G_n(H)) \text{ for } n \geq 1.$$

An *Generalized n-Superhyperstructure* is a pair:

$$\mathcal{GWSH}_n = (G_n(H), \diamond_n),$$

where:

- $G_n(H)$ is the n-th iteration of the generalized powerset operator G applied to H.
- $\diamond_n : G_n(H) \times G_n(H) \to \mathcal{P}(G_n(H)) \setminus \{\emptyset\}$ is a hyperoperation at the n-th level of generalization.

  This structure extends n-superhyperstructures by incorporating additional attributes, such as fuzziness, weighting, neutrosophic logic, or plithogenic properties, at each level of iteration.

Example 4.137 (Example of structures)**.** Below, we present several examples of such structures.

- *Generalized Fuzzy Hyperstructures:* Let $G(H)$ represent the fuzzy subsets of H. Then a Generalized Hyperstructure $(G(H), \diamond)$ might have $\diamond$ defined so that the combination of two fuzzy subsets results in a set of fuzzy subsets with certain combined membership functions.
- *Generalized Neutrosophic Hyperstructures:* If $G(H)$ denotes neutrosophic subsets of H, then $\diamond$ combines neutrosophic degrees of truth, indeterminacy, and falsity, yielding sets of neutrosophic subsets that capture more complex uncertainty at each operation.
- *Weighted Generalized Hyperstructures:* If $G(H)$ consists of subsets of H with weights or valuations, then a Generalized Hyperstructure uses $\diamond$ to combine these valuations in a hyperstructural manner.
- *Generalized n-Superhyperstructures:* Consider $G_n(H)$ where at the first level we have fuzzy sets, at the second level fuzzy sets of fuzzy sets, etc. The complexity grows, and so does the flexibility in modeling hierarchical and uncertain systems.

Theorem 4.138. *A Generalized Hyperstructure generalizes a Hyperstructure.*

Proof. A Hyperstructure $(S, \circ)$ is a special case of a Generalized Hyperstructure $(G(H), \diamond)$ when $G(H) = \mathcal{P}(H)$ (the ordinary powerset) and the hyperoperation $\diamond$ reduces to $\circ$. Thus, every hyperstructure is obtained by choosing a trivial generalization $G = \mathcal{P}$. Hence, Generalized Hyperstructures strictly generalize Hyperstructures. □

Theorem 4.139. *A Generalized n-Superhyperstructure generalizes a n-Superhyperstructure.*

Proof. This is evident. □

Theorem 4.140. *An n-Superhyperstructure is a special case of a Generalized n-Superhyperstructure.*

Proof. An n-Superhyperstructure $(\mathcal{P}^n(H), \star_n)$ uses the standard powerset operator $\mathcal{P}$. If we choose $G = \mathcal{P}$ as the generalized operator, then $G_n(H) = \mathcal{P}^n(H)$. By defining $\diamond_n$ to coincide with $\star_n$, we recover the original n-Superhyperstructure. Therefore, Generalized n-Superhyperstructures encompass n-Superhyperstructures as a special case. □

Theorem 4.141. *If the generalized operator G preserves certain algebraic or logical properties (e.g., closure under certain operations, monotonicity, or stability under fuzzy intersection/union), then the resulting Generalized n-Superhyperstructure inherits similar structural properties.*

Proof. Assume G is defined so that it integrates additional algebraic or logical properties into the subsets of H. For example, if $G(H)$ is closed under a particular fuzzy intersection operator and this closure extends to $G(G(H))$, and so forth. By induction, each $G_n(H)$ is constructed in a manner that preserves these properties at each level. Similarly, if $\diamond_n$ is defined to respect these properties, the entire Generalized n-Superhyperstructure will maintain the desired properties.

Thus, properties introduced by G are percolated through each iterative application, ensuring the resulting structure inherits the aimed attributes. □

Remark 4.142. The author believes that Generalized Hyperstructures and Generalized n-Superhyperstructures open a realm of possibilities. Future research in the following areas is highly anticipated.

- Integration with Complex Logics: Incorporating neutrosophic sets, plithogenic sets, or rough sets can model complex, uncertain systems.
- Hierarchical Modeling(cf. [49,286,335]): The n-th order generalization allows multi-layered hierarchical modeling, capturing different levels of abstraction.
- Applications in Decision Making(cf. [292, 338, 467]), AI, and Data Science (cf. [69, 397]): Handling complex attributes (like fuzziness or neutrality) within a hyperstructural framework can enhance decision-making models, neural network architectures, or data clustering algorithms.
- Combinational Explosion and Simplification Strategies: While generalization increases expressive power, it also increases complexity. Future research might focus on identifying simplifications, canonical forms, or embeddings into simpler structures, or establishing equivalences between different generalized frameworks.

4.12 Discussions: Application of Hyperstructures to Social Science and Other Domains

The concepts of hyperstructures and superhyperstructures can be applied beyond purely mathematical frameworks. This subsection explores their application to various concepts across different domains.

4.12.1 The Relationship Between Hyperstructures and Project Management

These concepts of Hyperstructure and n-Superhyperstructure can potentially be applied to Social Sciences as well. For instance, project management can be mathematically defined, albeit in an unconventional manner. Project management can be described as the structured planning, organization, and execution of tasks to achieve specific objectives within a defined timeframe and scope [41,142,168–170,287,350,447,461,474,510]. Generalized concepts such as program management [293,343,475] and portfolio management [97,111,285,402] are also well-known in this context.

While this approach may seem somewhat unconventional, it is possible to mathematically define these concepts and explore their relationships through the lens of Hyperstructures and n-Superhyperstructures. Relevant definitions and theorems are presented below.

Definition 4.143 (Project Management as a Structure)**.** Consider a non-empty finite set P of tasks (activities) that belong to a single project. A *Project Structure* is defined as a pair $(P, \star)$, where:

1. P is the set of tasks for one project.
2. $\star : P \times P \to P$ is a binary operation that composes tasks or results in an aggregated outcome (e.g., combining two tasks into a single aggregated deliverable).

The operation $\star$ is assumed to be *classical* (single-valued). This resembles a *classical structure* where each combination of tasks leads to a unique well-defined next step or integrated deliverable.

For example, if $P = \{t_1, t_2, \ldots, t_n\}$, then $(t_i \star t_j) \in P$ is a unique output task from merging or sequencing t_i and t_j.

Definition 4.144 (Program Management as a Hyperstructure)**.** A *Program* consists of multiple interrelated projects. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ be a collection of project task sets, each with its internal operation $\star$. Define a *Program Hyperstructure* as:

$$\mathcal{H} = (\mathcal{P}, \circ)$$

where $\circ : \mathcal{P} \times \mathcal{P} \to \mathcal{P}(\bigcup_i P_i)$ is a *hyperoperation*, mapping two project sets to a set of possible combined outcomes. Unlike the project-level operation $\star$, the program-level operation $\circ$ is *multi-valued*, reflecting the fact that combining tasks from different projects can lead to multiple possible integrated outcomes or sets of solutions.

For example, consider two projects P_a and P_b. The hyperoperation $\circ$ might produce:

$$P_a \circ P_b \subseteq \mathcal{P}(P_a \cup P_b),$$

a set of possible integrated workstreams or solution paths, rather than a single unique outcome.

Definition 4.145 (Portfolio Management as an n-Superhyperstructure)**.** A *Portfolio* encompasses multiple programs and possibly standalone projects, possibly arranged hierarchically. Let $Q = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_k\}$ be a finite set of programs, each a hyperstructure. To represent the complexity of a portfolio, we introduce iterative power set constructions leading to an (m, n)-SuperHyperOperation.

Define:

$$Q_n = \mathcal{P}^n(\bigcup_j \mathcal{H}_j)$$

as the n-th iterated powerset (or n-th super-level) of the union of all program task sets. An (m, n)-SuperHyperOperation $\diamond$ acts on these n-th level sets:

$$\diamond : (Q_n)^m \to \mathcal{P}_n^*(Q_n),$$

where $\mathcal{P}_n^*(\cdot)$ denotes an n-th order power set structure that can generate sets of sets of sets, and so forth.

The resulting structure:

$$(Q_n, \diamond)$$

is an *n-Superhyperstructure*, reflecting the multi-layered, hierarchical decision-making environment of a portfolio. Here, combining multiple programs (each already a hyperstructure) produces a richer, more complex structure, allowing multiple possible solutions at various hierarchical levels.

Theorem 4.146. *A program management structure (hyperstructure) generalizes a project management structure (classical structure).*

Proof. A project management structure $(P, \star)$ involves a single-valued operation on tasks. If we consider multiple projects $P_1, P_2, \ldots, P_m$ and define a hyperoperation $\circ$ that can produce multiple outcome sets from combining these P_i, we introduce multi-valuedness. This move from single-valued $\star$ to multi-valued $\circ$ is precisely the jump from a structure to a hyperstructure. Thus, a program (with multiple related projects) naturally corresponds to a hyperstructure generalization of a single project structure. □

Theorem 4.147. *Portfolio management, represented as an n-superhyperstructure, generalizes program management (hyperstructure) and project management (structure).*

Proof. A portfolio consists of multiple programs (hyperstructures). Introducing n-th power set constructions and n-SuperHyperOperations $\diamond$ on these program sets leads to an n-Superhyperstructure. For $n = 1$, we have a hyperstructure (program level). For $n > 1$, each iteration introduces additional layers of complexity and hierarchy, yielding an n-Superhyperstructure. Therefore, portfolio management, with its hierarchical combinations of programs and projects, is modeled by an n-Superhyperstructure that generalizes the hyperstructure (program) and the structure (project) cases. □

Remark 4.148. The analogy presented is metaphorical:

- *Project management* deals with a single organized set of tasks and decisions, analogous to a single-valued algebraic structure (classical structure).
- *Program management* deals with multiple projects, introducing branching sets of outcomes and integrated decision-making paths, analogous to a hyperstructure with multi-valued operations.
- *Portfolio management* involves an even higher-order hierarchical system of multiple programs and projects, where layering and complexity grows, analogous to an n-superhyperstructure.

4.12.2 Product Portfolio Management

Product Management refers to the process of managing individual products throughout their lifecycle, including strategy, development, and marketing [12, 105, 166, 334, 452]. Product Portfolio Management is the concept of managing overarching collections of products in a unified and strategic manner [109, 110, 112].

To formalize these concepts mathematically, we define Product Management (PM) and Product Portfolio Management (PPM) as follows.

Definition 4.149 (Product Management (PM))**.** Product Management focuses on the management of individual products. It is defined as:

$$PM = (P, R, M),$$

where:

- $P = \{p_1, p_2, \ldots, p_n\}$ is the set of products.
- $R \subseteq P \times P$ represents relationships between products, such as complementarity, competition, or dependency.
- $M : P \to \mathbb{R}^k$ is a management function that assigns attributes to each product, such as revenue, cost, or market share.

Example 4.150. Consider a company managing two products: a smartphone (p_1) and a tablet (p_2). The product set is:

$$P = \{p_1, p_2\}.$$

The relationship $R = \{(p_1, p_2)\}$ indicates that the smartphone and tablet are complementary. The management function M assigns attributes:

$$M(p_1) = (100, 50), \quad M(p_2) = (50, 30),$$

where $M(p_i)$ represents (revenue, cost) in million USD.

Definition 4.151 (Product Portfolio Management (PPM))**.** Product Portfolio Management manages collections of products as a whole. It is defined as:

$$PPM = (\mathcal{P}(P), R', M'),$$

where:

- $\mathcal{P}(P)$ is the power set of P, representing all possible subsets of products (i.e., product portfolios).
- $R' \subseteq \mathcal{P}(P) \times \mathcal{P}(P)$ represents relationships between product portfolios, such as competition or synergies.
- $M' : \mathcal{P}(P) \to \mathbb{R}^k$ is a portfolio management function that assigns attributes to portfolios, such as total revenue or strategic value.

Example 4.152. For the product set $P = \{p_1, p_2\}$, the power set is:

$$\mathcal{P}(P) = \{\emptyset, \{p_1\}, \{p_2\}, \{p_1, p_2\}\}.$$

A relationship $R' = \{(\{p_1\}, \{p_2\})\}$ indicates competition between the smartphone portfolio $\{p_1\}$ and the tablet portfolio $\{p_2\}$. The portfolio management function M' assigns total revenue:

$$M'(\{p_1, p_2\}) = 150 \quad \text{(million USD)}.$$

Definition 4.153 (Product n-SuperHyperPortfolio Management (Product SHPPM$_n$))**.** Product n-SuperHyperPortfolio Management generalizes Product Portfolio Management to include recursive layers of portfolios. It is defined as:

$$\text{SHPPM}_n = (\mathcal{P}_n(P), R^{(n)}, M^{(n)}),$$

where:

- $\mathcal{P}_n(P)$ is the n-th power set of P, recursively defined as:

$$\mathcal{P}_1(P) = \mathcal{P}(P), \quad \mathcal{P}_{k+1}(P) = \mathcal{P}(\mathcal{P}_k(P)) \text{ for } k \geq 1.$$

  This represents nested collections of portfolios.

- $R^{(n)} \subseteq \mathcal{P}_n(P) \times \mathcal{P}_n(P)$ represents relationships between elements of $\mathcal{P}_n(P)$, such as dependencies between hierarchical portfolios.
- $M^{(n)} : \mathcal{P}_n(P) \to \mathbb{R}^k$ is a management function that assigns attributes to elements of $\mathcal{P}_n(P)$, such as aggregated risks or strategic values.

Example 4.154. Let $P = \{p_1, p_2\}$. The first power set is:

$$\mathcal{P}_1(P) = \{\emptyset, \{p_1\}, \{p_2\}, \{p_1, p_2\}\}.$$

The second power set $\mathcal{P}_2(P)$ includes subsets of $\mathcal{P}_1(P)$. A relationship $R^{(2)} = \{(\{\{p_1\}, \{p_2\}\}, \{\{p_1, p_2\}\})\}$ reflects dependencies between portfolios and their sub-portfolios. The management function $M^{(2)}$ could assign aggregate risks:

$$M^{(2)}(\{\{p_1\}, \{p_2\}\}) = 0.5,$$

where 0.5 represents the aggregated risk of the nested portfolios.

In this paper, we introduced Project Portfolio Management and Product Portfolio Management. However, it is important to note that portfolio management is not limited to projects or products; it is a widely studied concept across various contexts. For instance, Active Portfolio Management [83,84,201,212], Agile Portfolio Management [298,307,439,514], Investment Portfolio Management [42,379], Business Portfolio Management [25,466], Passive Portfolio Management [316,515], and Strategic Portfolio Management [23,34] are all areas of research within this domain. These concepts can also be examined as hyperstructures and superhyperstructures using similar methodologies. Further research into these areas is anticipated as needed.

4.12.3 Programming Function Hyperstructure

Programming is the process of designing, writing, testing, and maintaining code to create functional software applications [257, 417]. A Programming Function is a reusable computational unit in programming, mapping inputs to outputs based on defined operations or logic (cf. [342, 546, 581]). It can be analyzed using hyperstructure and superhyperstructure frameworks. The definitions and related concepts are detailed below.

Definition 4.155 (Programming Function Hyperstructure)**.** Let $S = \{f_1, f_2, \ldots, f_n\}$ be a set of functions within a programming context, where each f_i represents an individual function or a method. The *Programming Function Hyperstructure* is defined as:

$$\mathcal{H}_P = (\mathcal{P}(S), \circ),$$

where:

- $\mathcal{P}(S)$: The powerset of S, representing all possible subsets of functions, including individual functions and their combinations.
- $\circ$: A binary operation defined on subsets of $\mathcal{P}(S)$, such as function composition, union, or a custom-defined operation specific to the programming framework.

Example 4.156 (Hyperstructure in Programming)**.** Consider a set of programming functions $S = \{f_1, f_2, f_3\}$, where:

- $f_1(x) = x + 1$: A function that increments its input by 1.
- $f_2(x) = 2x$: A function that doubles its input.
- $f_3(x) = x^2$: A function that squares its input.

The powerset $\mathcal{P}(S)$ includes:

$$\mathcal{P}(S) = \{\emptyset, \{f_1\}, \{f_2\}, \{f_3\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}, \{f_1, f_2, f_3\}\}.$$

Define an operation $\circ$ as the sequential composition of functions. For example:

$$\{f_1, f_2\} \circ \{f_3\} = \{f_3 \circ f_1, f_3 \circ f_2\}.$$

If $f_1(x) = x + 1$ and $f_3(x) = x^2$, then:

$$f_3 \circ f_1(x) = (x + 1)^2.$$

Theorem 4.157. *The Programming Function Hyperstructure generalizes function organization in software systems, enabling structured analysis of combinations, dependencies, and compositions within codebases.*

Proof. By definition, $\mathcal{P}(S)$ provides all possible groupings of functions. The operation $\circ$ defines interactions between these groupings, thereby encapsulating the structural and behavioral complexity of a software system in a hyperstructure framework. □

Definition 4.158 (Programming Function n-SuperHyperstructure)**.** Let $S = \{f_1, f_2, \ldots, f_n\}$ be a set of functions within a programming context. The *Programming Function n-SuperHyperstructure* is defined as:

$$\mathcal{SH}_n = (\mathcal{P}_n(S), \circ^{(n)}),$$

where:

- $\mathcal{P}_n(S)$: The n-th powerset of S, defined recursively as:

  $$\mathcal{P}_1(S) = \mathcal{P}(S), \quad \mathcal{P}_{k+1}(S) = \mathcal{P}(\mathcal{P}_k(S)) \quad \text{for } k \geq 1.$$

  This represents all n-level combinations of subsets of S.
- $\circ^{(n)}$: A generalized n-ary operation defined on $\mathcal{P}_n(S)$, such as nested function compositions, unions, or other operations specific to the programming framework.

Example 4.159 (n-SuperHyperstructure in Programming)**.** Let $S = \{f_1, f_2, f_3\}$, where:

- $f_1(x) = x + 1$: A function that increments its input by 1.

- $f_2(x) = 2x$: A function that doubles its input.
- $f_3(x) = x^2$: A function that squares its input.

The first powerset is:

$$\mathcal{P}_1(S) = \mathcal{P}(S) = \{\emptyset, \{f_1\}, \{f_2\}, \{f_3\}, \{f_1, f_2\}, \{f_1, f_3\}, \{f_2, f_3\}, \{f_1, f_2, f_3\}\}.$$

The second powerset $\mathcal{P}_2(S)$ is the powerset of $\mathcal{P}_1(S)$, containing all subsets of $\mathcal{P}_1(S)$.

Define $\circ^{(2)}$ as the nested composition of subsets of functions. For instance:

$$\{\{f_1\}, \{f_2\}\} \circ^{(2)} \{\{f_3\}\} = \{\{f_3 \circ f_1\}, \{f_3 \circ f_2\}\}.$$

If $f_1(x) = x + 1$ and $f_3(x) = x^2$, then:

$$f_3 \circ f_1(x) = (x + 1)^2.$$

Similarly, $\mathcal{P}_3(S)$ extends this to a third level, where operations are defined on subsets of $\mathcal{P}_2(S)$.

Theorem 4.160. *The n-SuperHyperstructure provides a hierarchical framework for analyzing multi-level relationships and operations within programming function systems.*

Proof. By recursively constructing $\mathcal{P}_n(S)$, the n-SuperHyperstructure models higher-order dependencies and combinations of functions. The operation $\circ^{(n)}$ enables analysis and optimization at each hierarchical level, making it a powerful tool for understanding complex program structures. □

4.12.4 Software Framework Ecosystem

To enhance the efficiency of programming and development, numerous programming libraries and software frameworks have been introduced (e.g. [26, 37, 118, 231, 310, 401]). This section attempts to represent these concepts using hyperstructure and superhyperstructure frameworks and mathematically define the *Software Framework Ecosystem*. It is important to note that this definition is conceptual and may be refined further, as more comprehensive definitions could emerge in the future.

Definition 4.161 (Programming Library as a Structure)**.** A *Programming Library* is a reusable collection of functions and routines designed to perform specific tasks in a programming environment. Formally, a library can be defined as:

$$\mathcal{L} = (L, \circ),$$

where:

- $L = \{f_1, f_2, \ldots, f_n\}$: A set of functions or routines provided by the library.
- $\circ$: An operation that combines or sequences functions within the library, such as function composition.

Example 4.162. Consider `NumPy` [229, 384], a Python library for numerical computations. The functions $L = \{\text{dot}, \text{sum}, \text{mean}\}$ represent specific tasks like matrix multiplication, summation, and averaging. The operation $\circ$ combines these functions, for example:

$$\text{result} = \text{mean}(\text{sum}(\text{dot}(A, B))),$$

where A and B are input matrices.

Definition 4.163 (Programming Framework as a Hyperstructure)**.** A *Programming Framework* is a structured environment that integrates multiple libraries and defines their interactions to support the development of complex systems (cf. [46, 158, 159, 415]). It can be represented as:

$$\mathcal{F} = (\mathcal{P}(L), \circ),$$

where:

- $\mathcal{P}(L)$: The powerset of libraries $L = \{L_1, L_2, \ldots, L_m\}$, where each L_i is a programming library.
- $\circ$: A hyperoperation that models the interaction or data flow between libraries, such as API calls or dependency relationships.

Theorem 4.164. *A Programming Framework possesses the structure of a hyperstructure.*

Proof. This follows directly from the definition. □

Example 4.165. The Django framework (cf. [161, 450]) integrates libraries for handling requests, databases, and templates:

$$L = \{\text{views.py}, \text{models.py}, \text{templates/}\}.$$

The operation $\circ$ models their interaction, such as routing user requests (`views.py`) to database operations (`models.py`) and rendering templates (`templates/`).

Definition 4.166 (Framework Ecosystem as a Superhyperstructure)**.** A *Framework Ecosystem* (cf. [2]) is a higher-order system composed of multiple frameworks interacting in a hierarchical or networked structure. It is formally defined as:

$$\mathcal{SF} = (\mathcal{P}_n(F), \circ),$$

where:

- $\mathcal{P}_n(F)$: The n-th powerset of frameworks $F = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_p\}$, capturing higher-level relationships between frameworks.
- $\circ$: A superhyperoperation that models interactions, such as API integration, data exchange, or orchestration between frameworks.

Theorem 4.167. *A Framework Ecosystem possesses the structure of a superhyperstructure.*

Proof. This follows directly from the definition. □

Example 4.168. A full-stack web application (cf. [7, 404]) combines Django (backend) and React (frontend) [149]:

$$F = \{\mathcal{F}_{\text{Django}}, \mathcal{F}_{\text{React}}\}.$$

The interaction $\circ$ is modeled as API calls, such as Django exposing REST endpoints consumed by React. The ecosystem is represented as:

$$\mathcal{SF} = (\mathcal{P}_n(F), \circ),$$

where multiple frameworks interact to build a cohesive application.

4.12.5 Educational Curriculum and Courses

Educational Curriculum is a structured framework outlining courses, subjects, and learning objectives to guide and assess educational progress [217, 228, 329]. Educational Curriculum and Courses are examined using hyperstructure and superhyperstructure frameworks.

Definition 4.169 (Structure: Individual Course or Subject)**.** A course C is modeled as a tuple:

$$C = (T, P, O),$$

where:

- T: The title of the course (e.g., "Mathematics 101").
- P: Prerequisites for the course.
- O: Learning outcomes or objectives.

Definition 4.170 (Hyperstructure: Course Collection and Relationships)**.** The curriculum, representing courses and their interrelations, is defined as a hyperstructure:

$$\mathcal{H}_C = (\mathcal{P}(C), \circ),$$

where:

- $\mathcal{P}(C)$: The powerset of all courses.
- $\circ$: Operations modeling relationships such as prerequisites, co-requisites, or course progression paths.

Definition 4.171 (Superhyperstructure: Hierarchical Course Organization)**.** The curriculum extended across programs or institutions is represented as:

$$\mathcal{SH}_C^n = (\mathcal{P}_n(C), \circ),$$

where:

- $\mathcal{P}_n(C)$: The n-th powerset of courses, capturing nested structures (e.g., programs, institutions).
- $\circ$: Operations modeling collaborative or hierarchical relationships among courses.

4.12.6 Products and Components

Products and Components are analyzed and extended using hyperstructure and superhyperstructure frameworks.

Definition 4.172 (Structure: Individual Product or Component)**.** A component P is defined as:

$$P = (ID, F, C),$$

where:

- ID: A unique identifier for the component (e.g., "Engine").
- F: A set of features or specifications.
- C: Constraints such as compatibility or size.

Definition 4.173 (Hyperstructure: Component Collection and Interactions)**.** The system of components is represented as:

$$\mathcal{H}_P = (\mathcal{P}(P), \circ),$$

where:

- $\mathcal{P}(P)$: The powerset of components.
- $\circ$: Operations describing assembly rules or interactions between components.

Definition 4.174 (Superhyperstructure: Hierarchical Component Organization)**.** A nested system of products and subsystems is represented as:

$$\mathcal{SH}_P^n = (\mathcal{P}_n(P), \circ),$$

where:

- $\mathcal{P}_n(P)$: The n-th powerset of components, capturing nested subsystems or inter-product (cf. [579]) shared components.
- $\circ$: Operations describing hierarchical dependencies across products.

4.12.7 Economic Systems and Transactions

Economic Systems are frameworks for organizing production, distribution, and consumption of goods and services, encompassing resources, institutions, and interactions [163, 196, 241, 320, 391]. This concept is extended using hyperstructure and superhyperstructure to model complex economic relationships and hierarchical interactions.

Definition 4.175 (Structure: Individual Transaction)**.** A transaction T is modeled as:

$$T = (A, B, X, P),$$

where:

- A: Buyer.
- B: Seller.
- X: Item or service exchanged.
- P: Price or value.

Example 4.176 (Structure: Individual Transaction)**.** An individual economic transaction T can be modeled as:

$$T = (A, B, X, P),$$

where:

- A: The buyer involved in the transaction.
- B: The seller providing goods or services.
- X: The item or service exchanged in the transaction.
- P: The price or value assigned to the transaction.

For instance, consider a transaction where a consumer buys a book from a bookstore for \$20. Here, A is the consumer, B is the bookstore, X is the book, and $P = 20$.

Definition 4.177 (Hyperstructure: Transaction Collection and Dependencies)**.** The market, representing all transactions and their interdependencies, is defined as:

$$\mathcal{H}_T = (\mathcal{P}(T), \circ),$$

where:

- $\mathcal{P}(T)$: The powerset of transactions.
- $\circ$: Operations describing dependencies such as supply-demand relationships or price dynamics.

Example 4.178 (Hyperstructure: Transaction Collection and Dependencies)**.** The market, encompassing all transactions and their interdependencies, can be represented as a hyperstructure:

$$\mathcal{H}_T = (\mathcal{P}(T), \circ),$$

where:

- $\mathcal{P}(T)$: The powerset of transactions T, representing all subsets of transactions in the market.
- $\circ$: Operations that describe dependencies such as supply-demand relationships, price adjustments, or the impact of one transaction on another.

Consider a specific example in the retail market. Suppose T includes transactions such as:

$$T = \{T_1, T_2, T_3\},$$

where:

- T_1: Customer purchases a smartphone.
- T_2: Retailer orders new stock of smartphones from a supplier.
- T_3: Supplier orders components from a chip manufacturer.

The powerset $\mathcal{P}(T)$ represents all combinations of these transactions, such as $\{T_1, T_2\}$, indicating that customer demand leads the retailer to order new stock.

An operation $\circ$ could represent the dependency:

$$T_1 \circ T_2 = \text{Increased stock order due to customer purchases.}$$

Similarly:

$$T_2 \circ T_3 = \text{Supplier increases chip orders due to retailer's demand.}$$

This hyperstructure captures how a single customer purchase propagates through the supply chain, influencing subsequent transactions.

Definition 4.179 (Superhyperstructure: Hierarchical Market Interactions)**.** The global economic system [545] is represented as:

$$\mathcal{SH}_T^n = (\mathcal{P}_n(T), \circ),$$

where:

- $\mathcal{P}_n(T)$: The n-th powerset of transactions, modeling interactions across markets or economies.
- $\circ$: Operations capturing inter-market dynamics or global supply chains.

Example 4.180 (Superhyperstructure: Hierarchical Market Interactions)**.** The global economic system, capturing hierarchical interactions across markets or economies, is represented as a superhyperstructure:

$$\mathcal{SH}_T^n = (\mathcal{P}_n(T), \circ),$$

where:

- $\mathcal{P}_n(T)$: The n-th powerset of transactions, representing interactions across multiple levels, such as regional, national, and global markets.
- $\circ$: Operations that capture inter-market dynamics, global supply chains, or international trade relations.

For example, consider T containing transactions at different levels:

$$T = \{T_1, T_2, T_3, T_4\},$$

where:

- T_1: Local farmer sells produce to a regional distributor.
- T_2: Distributor sells produce to a national retailer.
- T_3: Retailer exports produce to an international market.
- T_4: International market supplies value-added goods back to the retailer.

The n-th powerset $\mathcal{P}_n(T)$ represents combinations such as:

$$\mathcal{P}_2(T) = \{\{T_1, T_2\}, \{T_3, T_4\}, \{T_1, T_3, T_4\}\}.$$

An operation $\circ$ could model how these interactions influence global trade:

$$\{T_1, T_2\} \circ \{T_3, T_4\} = \text{Price adjustments based on export-import balance.}$$

This superhyperstructure describes how local economic activities (e.g., farming) cascade into regional and global economic systems, illustrating complex market hierarchies.

4.12.8 Musical Patterns and Relationships

This subsection explores the representation of Musical Patterns and Relationships using hyperstructures and superhyperstructures.

Definition 4.181 (Structure: Individual Note or Phrase)**.** A note or phrase M is modeled as:

$$M = (N, D, I),$$

where:

- N: The pitch or note (e.g., "C4") (cf. [238, 309]).
- D: Duration (e.g., "quarter note") (cf. [519]).
- I: Intensity (e.g., "forte").

Definition 4.182 (Hyperstructure: Musical Patterns and Relationships)**.** A musical composition is represented as:

$$\mathcal{H}_M = (\mathcal{P}(M), \circ),$$

where:

- $\mathcal{P}(M)$: The powerset of notes or phrases.
- $\circ$: Operations describing relationships such as harmony, rhythm, or counterpoint.

Example 4.183 (Hyperstructure: Musical Patterns and Relationships)**.** A musical composition (cf. [80, 556]) consisting of patterns and relationships between notes is modeled as:

$$\mathcal{H}_M = (\mathcal{P}(M), \circ),$$

where:

- $\mathcal{P}(M)$: The powerset of notes M, representing all possible combinations of notes or phrases.
- $\circ$: Operations that define relationships between notes, such as harmonic progression, rhythmic patterns, or melodic counterpoint.

For example:

- Let $M = \{$"C4 quarter forte", "E4 quarter piano", "G4 half mezzo-forte"$\}$.
- The powerset $\mathcal{P}(M)$ includes subsets such as $\{$"C4 quarter forte"$\}$ or $\{$"C4 quarter forte", "E4 quarter piano"$\}$.

- An operation $\circ$ could represent harmony, e.g.,

$$\{\text{"C4 quarter forte", "E4 quarter piano"}\} \circ \{\text{"G4 half mezzo-forte"}\} = \text{C-major chord}.$$

Definition 4.184 (Superhyperstructure: Hierarchical Music Organization)**.** A complex musical piece with nested arrangements is defined as:

$$\mathcal{SH}_M^n = (\mathcal{P}_n(M), \circ),$$

where:

- $\mathcal{P}_n(M)$: The n-th powerset of notes, capturing sections (e.g., phrases $\to$ movements $\to$ orchestration).
- $\circ$: Operations modeling interdependencies between movements or instruments.

Example 4.185 (Superhyperstructure: Hierarchical Music Organization)**.** A symphony with multiple movements (cf. [374]) and nested arrangements is modeled as:

$$\mathcal{SH}_M^n = (\mathcal{P}_n(M), \circ),$$

where:

- $\mathcal{P}_n(M)$: The n-th powerset of notes M, capturing hierarchical structures such as phrases, movements, and orchestration.
- $\circ$: Operations modeling dependencies, e.g., how phrases within movements interact or how instrument sections collaborate.

For example:

- M: Individual notes from a violin section, e.g., {"C4", "E4", "G4"}.
- $\mathcal{P}_1(M)$: Represents phrases in the violin section.
- $\mathcal{P}_2(M)$: Represents the interaction of violin phrases with cello phrases.
- $\circ$: Describes interdependencies, e.g.,

$$(\mathcal{P}_1(\text{Violin})) \circ (\mathcal{P}_1(\text{Cello})) = \text{String harmony}.$$

This framework can capture complex orchestral compositions, such as how a flute melody in one movement transitions into a violin counterpoint in another.

4.12.9 Company's Organizational Structure

Organizational structure has been widely discussed in various contexts (cf. [33, 104, 131, 227, 259, 301, 323, 340, 426, 525]). For example, a company's organizational structure [256] can be effectively represented using a superhyperstructure. Here, divisions (departments or business units [214, 232]) are considered as nodes in a hierarchical framework, with each layer introducing new complexities, dependencies, and relationships.

Definition 4.186 (Division Structure)**.** (cf. [327]) Let $D = \{d_1, d_2, \dots, d_n\}$ represent the set of all divisions in the company. Each division may have sub-divisions, forming a hierarchical dependency structure.

Definition 4.187 (Relations and Constraints)**.** Define $R \subseteq D \times D$, where $(d_i, d_j) \in R$ indicates that d_i depends on d_j (e.g., for resources, deliverables, or strategy alignment).

Definition 4.188 (Hyperstructure for Interactions)**.** A hyperstructure is defined to capture multiple interactions, such as resource sharing across divisions:

$$\mathcal{H}(D) = \{H \mid H \subseteq D, |H| > 1\},$$

where H represents a subset of divisions collaborating on a common objective.

Theorem 4.189. *The hierarchical structure of a company's divisions can be represented as a n-superhyperstructure.*

Proof. 1. *Base Case ($n = 1$):* At the first level, the company's divisions form a hyperstructure $\mathcal{H}(D)$, where subsets $H \subseteq D$ represent interacting divisions.

2. *Induction Hypothesis:* Assume that for k, the structure up to k-th level can be represented as $\mathcal{P}_k(D)$, capturing dependencies among divisions and sub-divisions recursively.

3. *Induction Step ($n = k + 1$):* At the $k + 1$-th level, meta-interactions (e.g., global strategy alignment across divisions) introduce dependencies among subsets of subsets, formalized by $\mathcal{P}_{k+1}(D)$.

Thus, by induction, the company's structure can be fully captured as a n-superhyperstructure. □

Example 4.190. Consider the hierarchical structure of a multinational corporation:

- *Level $n = 1$: Divisions.* Divisions are grouped by function:

$$D = \{\text{Marketing}, \text{Sales}, \text{Engineering}, \text{Operations}\}.$$

- *Level $n = 2$: Interactions Between Subgroups.* Interactions between divisions can be represented as:

$$\mathcal{P}_2(D) = \{\{\text{Marketing, Sales}\}, \{\text{Engineering, Operations}\}\}.$$

- *Level $n = 3$: Global Coordination.* Coordination between interaction groups can be expressed as:

$$\mathcal{P}_3(D) = \{\{\{\text{Marketing, Sales}\}, \{\text{Engineering, Operations}\}\}\}.$$

- *Level $n = 4$: Corporate Strategy.* Top-level dependencies are captured, aligning global resources and strategy.

From the above discussion, the Organizational SuperHyperstructure can be mathematically defined as follows.

Definition 4.191 (Organizational SuperHyperstructure with Superedges)**.** An *Organizational SuperHyperstructure* is a hierarchical framework representing a company's organizational structure, defined as a tuple:

$$O = (L, \mathcal{N}, \mathcal{S}),$$

where:

- $L = \{L_1, L_2, \dots, L_n\}$ represents the hierarchical levels of the organization, with L_1 being the top level (e.g., executive board) and L_n the lowest level (e.g., operational teams).
- $\mathcal{N} = \bigcup_{i=1}^{n} \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of nodes (departments, divisions, teams, or individuals) at level L_i.
- $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m\}$ represents the set of superedges. Each superedge connects a subset of nodes from potentially multiple levels, forming a multi-layered relationship or dependency.

Each node $N \in \mathcal{N}_i$ may have attributes or roles $A(N) = \{a_1, a_2, \dots, a_k\}$, capturing its functions or responsibilities. Each superedge $S \in \mathcal{S}$ can be represented as:

$$S \subseteq \bigcup_{i=1}^{n} \mathcal{P}^m(\mathcal{N}_i),$$

where $\mathcal{P}^m(\mathcal{N}_i)$ denotes the m-th power set of nodes at level L_i, allowing for multi-layer and multi-node connections.

A SuperHyperstructure emerges when L, $\mathcal{N}$, or $\mathcal{S}$ incorporate the following:

- *Nested Relationships:* Recursive dependencies where higher-level decisions affect lower-level actions.
- *Plithogenic Attributes:* Overlapping or conflicting roles of nodes, modeled with fuzzy or neutrosophic sets.
- *Superedges:* Higher-dimensional relationships that span multiple levels or subsets of nodes, enabling intricate collaborations and dependencies.

Example 4.192 (Organizational SuperHyperstructure with Superedges)**.** Consider a multinational corporation structured as follows:

- At L_1 (Executive Level): The board of directors sets global strategies, such as sustainability goals or profit targets.
- At L_2 (Regional Level): Regional managers adapt these strategies to local markets, allocating resources based on regional needs.
- At L_3 (Department Level): Departments like marketing, R&D, and logistics create localized plans.
- At L_4 (Team Level): Teams execute specific tasks, such as running ad campaigns or managing inventory.

Superedges in this structure:

- A superedge at S_1 connects (L_1, L_3), representing the direct influence of executive decisions on departmental goals.
- A superedge at S_2 spans (L_2, L_4), capturing the flow of resource allocation from regional managers to operational teams.
- A multi-level superedge at S_3 includes nodes from (L_1, L_2, L_3), describing collaborative efforts across global, regional, and departmental levels for a sustainability initiative.

These superedges enable the representation of intricate, multi-layered relationships within the company, supporting advanced analysis of organizational dependencies and strategies.

The above concept can be applied to various frameworks, such as government structures (Government Management) [361, 448], legal structures [144], product management [81, 116], software structures [153, 414], and school structures (School Management) [89, 356, 380]. The author hopes that future research will further explore the application of these concepts across various fields.

4.13 Discussions: HyperGame Theory

Game Theory is a mathematical framework for analyzing strategic interactions among rational decision-makers in competitive or cooperative scenarios [50, 91, 141, 368, 377, 389]. An extension of this, HyperGame Theory, has been recently defined [56, 57, 248, 303, 529, 583]. HyperGame Theory can, in certain respects, be viewed as possessing the structure of a Hyperstructure. Furthermore, a more generalized concept, the superhypergame, is defined as follows. Future research on these topics is expected to advance further.

Definition 4.193 (Game)**.** (cf. [141, 389]) A *Game* is a formal model of strategic interaction among a finite set of players $N = \{1, 2, \dots, n\}$. It is defined as:

$$G = (N, S, U),$$

where:

- N is the set of players.
- $S = \{S_1, S_2, \dots, S_n\}$ is the set of strategy spaces, where S_i is the set of strategies available to player i.

- $U = \{U_1, U_2, \ldots, U_n\}$ is the set of utility functions, where $U_i : S_1 \times S_2 \times \cdots \times S_n \to \mathbb{R}$ assigns a payoff to player i for each strategy profile $(s_1, s_2, \ldots, s_n)$.

Example 4.194 (Battle of the Sexes)**.** Consider two players $N = \{1, 2\}$ with the following scenario:

- $S_1 = S_2 = \{\text{Opera (O)}, \text{Football (F)}\}$, where player 1 prefers Opera and player 2 prefers Football.
- The payoff matrix is as follows:

	O	F
O	$(2, 1)$	$(0, 0)$
F	$(0, 0)$	$(1, 2)$

In this game, both players aim to coordinate but have conflicting preferences, leading to two pure strategy Nash equilibria: (O, O) and (F, F).

Definition 4.195 (HyperGame)**.** (cf. [56]) A *HyperGame* generalizes a game to account for players' differing perceptions of the game structure, strategies, or payoffs. It is defined as:

$$H = \{G_1, G_2, \ldots, G_n\},$$

where:

- $G_i = (N, S_i, U_i)$ is the perceived game of player i, where:
  - $S_i = \{S_{i1}, S_{i2}, \ldots, S_{in}\}$ is the set of strategies as perceived by i.
  - $U_i = \{U_{i1}, U_{i2}, \ldots, U_{in}\}$ is the set of utility functions as perceived by i.

Example 4.196 (Perceived Asymmetric Payoffs)**.** Suppose two players $N = \{1, 2\}$ face a situation where their perceptions of the game differ:

- Player 1 perceives the game G_1 as:

$$G_1 = \begin{array}{c|c|c} & \text{O} & \text{F} \\ \hline \text{O} & (2,1) & (0,0) \\ \text{F} & (0,0) & (1,2) \end{array}$$

- Player 2 perceives the game G_2 differently, assuming Player 1 values coordination more than their own preference:

$$G_2 = \begin{array}{c|c|c} & \text{O} & \text{F} \\ \hline \text{O} & (3,3) & (0,0) \\ \text{F} & (0,0) & (1,2) \end{array}$$

Here, Player 1's actual preference differs from Player 2's perception, illustrating a HyperGame.

Theorem 4.197. *A HyperGame $H = \{G_1, G_2, \ldots, G_n\}$, where $G_i = (N, S_i, U_i)$ represents the game perceived by player i, is a generalization of a classical Game $G = (N, S, U)$.*

Proof. To demonstrate this theorem, it is necessary to show that any classical Game G can be represented as a special case of a HyperGame H where all players share the same perception of the game.

First, consider the structure of a classical Game G. It is defined as a tuple $G = (N, S, U)$, where $N = \{1, 2, \ldots, n\}$ is the set of players, $S = \{S_1, S_2, \ldots, S_n\}$ is the set of strategy spaces, and $U = \{U_1, U_2, \ldots, U_n\}$ is the set of utility functions. In a classical Game, it is assumed that all players have a common and complete understanding of the game, meaning that $S_i = S$ and $U_i = U$ for all players $i \in N$.

Now consider the structure of a HyperGame H. A HyperGame is defined as $H = \{G_1, G_2, \ldots, G_n\}$, where each $G_i = (N, S_i, U_i)$ represents the game as perceived by player i. Here, $S_i = \{S_{i1}, S_{i2}, \ldots, S_{in}\}$ is the set of strategies as perceived by player i, and $U_i = \{U_{i1}, U_{i2}, \ldots, U_{in}\}$ is the set of utility functions as perceived by player i. The key difference between a classical Game and a HyperGame lies in the fact that the perceptions of the game can vary across players in a HyperGame.

If all players in a HyperGame share identical perceptions of the game, it follows that for all $i \in N$, $S_i = S$ and $U_i = U$. In this scenario, the perceived games G_i are identical for all players. Consequently, the HyperGame H simplifies to $H = \{G, G, \ldots, G\}$, which is equivalent to $H = \{G\}$. This demonstrates that a classical Game G is a special case of a HyperGame H where there are no differences in players' perceptions.

In a more general HyperGame, S_i and U_i can vary between players, allowing for the modeling of scenarios involving incomplete information, misperceptions, or asymmetric knowledge among the players. These characteristics extend the applicability of HyperGame beyond that of classical Game theory.

Since any classical Game G can be expressed as a specific instance of a HyperGame H, where all players have identical perceptions, it is clear that a HyperGame generalizes the concept of a classical Game. □

Theorem 4.198. *A HyperGame $H = \{G_1, G_2, \ldots, G_n\}$, where each $G_i = (N, S_i, U_i)$ represents the perceived game of player i, possesses the mathematical structure of a Hyperstructure.*

Proof. To prove this, we verify that a HyperGame satisfies the formal definition of a Hyperstructure, which is given as:

$$\mathcal{H} = (\mathcal{P}(S), \circ),$$

where $\mathcal{P}(S)$ is the powerset of a base set S, and $\circ$ is an operation defined on elements of $\mathcal{P}(S)$.

Let $S = \bigcup_{i=1}^{n} S_i$ be the union of all strategy spaces S_i perceived by players $i \in N$. Since $\mathcal{P}(S)$ is the set of all subsets of S, it includes every subset of the strategies available in the HyperGame.

The perceived games G_i are defined by subsets of $\mathcal{P}(S)$, specifically $S_i \subseteq S$, and thus $H = \{G_1, G_2, \ldots, G_n\}$ naturally resides within $\mathcal{P}(S)$. This satisfies the requirement that the HyperGame is built on the powerset of a base set.

In the context of a HyperGame, the operation $\circ$ represents the combination of strategy profiles across players' perceptions. Formally, $\circ$ is a mapping:

$$\circ : \mathcal{P}(S_1) \times \mathcal{P}(S_2) \to \mathcal{P}(S),$$

where $\circ(A, B) = \{(a, b) \mid a \in A, b \in B\}$, and $A \subseteq S_1, B \subseteq S_2$. This operation is closed within $\mathcal{P}(S)$ because any combination of subsets A, B from $\mathcal{P}(S)$ produces another subset of S.

To verify that $\circ$ satisfies the requirements of a Hyperstructure:

- *Closure:* For any $A, B \in \mathcal{P}(S)$, $\circ(A, B) \subseteq \mathcal{P}(S)$, since the Cartesian product of subsets is also a subset of $\mathcal{P}(S)$.
- *Associativity:* For $A, B, C \in \mathcal{P}(S)$,

  $$\circ(\circ(A, B), C) = \circ(A, \circ(B, C)),$$

  which holds because the Cartesian product operation is associative.
- *Identity:* The empty set $\emptyset \in \mathcal{P}(S)$ serves as an identity element:

  $$\circ(A, \emptyset) = A, \quad \circ(\emptyset, B) = B.$$

The HyperGame $H = \{G_1, G_2, \ldots, G_n\}$ is formed by the interaction of players' perceptions, where each $G_i = (N, S_i, U_i)$ is based on subsets of $\mathcal{P}(S)$. The operation $\circ$ combines strategy spaces S_i across perceptions, ensuring that H satisfies the recursive structure of a Hyperstructure.

Since the HyperGame H satisfies the powerset-based structure, supports a closed operation $\circ$, and adheres to the properties of closure, associativity, and identity, we conclude that:

$$H \text{ possesses the structure of a Hyperstructure.}$$

□

Definition 4.199 (n-SuperHyperGame)**.** An *n-SuperHyperGame* extends a HyperGame by including recursive levels of perception, where each player perceives other players' perceptions up to n-levels. It is defined as:

$$SH_n = \{H_1, H_2, \ldots, H_n\},$$

where:

- $H_k = \{G_{1k}, G_{2k}, \ldots, G_{nk}\}$ is the k-th level HyperGame, representing each player's k-th level perception.
- $G_{ik} = (N, S_{ik}, U_{ik})$, where:
  - S_{ik} is the strategy space at level k as perceived by player i.
  - U_{ik} is the utility function at level k as perceived by player i.

Example 4.200 (Two-level SuperHyperGame: Battle of Misaligned Beliefs)**.** Consider $N = \{1, 2\}$, where at level 1:

- Player 1 perceives G_1 as:

	O	F
O	$(2, 1)$	$(0, 0)$
F	$(0, 0)$	$(1, 2)$

- Player 2 perceives G_2 as:

	O	F
O	$(3, 3)$	$(0, 0)$
F	$(0, 0)$	$(1, 2)$

At level 2:

- Player 1 believes Player 2's perception is G_2.
- Player 2 believes Player 1's perception is G_1.

Thus, the 2-SuperHyperGame is represented as:

$$SH_2 = \{\{G_1, G_2\}, \{G_1, G_2\}\}.$$

This recursive structure highlights the complexity introduced by differing beliefs about perceptions.

Theorem 4.201. *An n-SuperHyperGame generalizes a HyperGame.*

Proof. This is evident. The proof follows the same reasoning as in the case of a HyperGame. □

Theorem 4.202. *An n-SuperHyperGame possesses the structure of a superhyperstructure.*

Proof. This is evident. The proof follows the same reasoning as in the case of a HyperGame. □

4.14 Discussions: Hyperdecision-making

Decision-making is the process of selecting the best option from alternatives based on criteria, constraints, and desired outcomes (cf. [92, 115, 146, 341, 509, 544]). Related theories, such as Social Choice Theory [38, 77, 318, 390, 465], are also well-known and have been extensively studied, similar to Decision-making. This concept is extended using Hyperstructures and Superhyperstructures, which allow for the representation of decision-making frameworks where higher-level or preceding decisions influence lower-level or subsequent decisions.

Definition 4.203 (Decision-Making)**.** (cf. [92, 115, 146, 341, 509, 544]) Decision-making is the process of selecting an optimal choice from a set of alternatives $A = \{a_1, a_2, \dots, a_n\}$ under given constraints $C = \{c_1, c_2, \dots, c_m\}$ and criteria $K = \{k_1, k_2, \dots, k_p\}$. Formally, it can be represented as:

$$a^* = \arg\max_{a \in A} \mathcal{U}(a, C, K),$$

where $\mathcal{U} : A \times C \times K \to \mathbb{R}$ is a utility function quantifying the desirability of each alternative a given the constraints and criteria.

Example 4.204 (Personal Investment Decision)**.** (cf. [65, 353]) A person decides how to allocate an amount I among n investment options $A = \{a_1, a_2, \dots, a_n\}$, such as stocks, bonds, or savings accounts. The constraints include:

- A maximum risk tolerance c_1: $\mathrm{Risk}(a_i) \leq c_1$ for all i.
- A minimum liquidity requirement c_2: $\mathrm{Liquidity}(a_i) \geq c_2$.

The criteria involve maximizing expected returns:

$$a^* = \arg\max_{a \in A} \mathbb{E}[\mathrm{Returns}(a)].$$

Definition 4.205 (Hyperdecision-making)**.** Hyperdecision-making refers to a decision-making process where individuals or systems face an *overabundance of choices* or alternatives, often characterized by high-dimensional relationships or dependencies. It incorporates:

- *Choice Overload:* Situations with an excessive number of options, leading to decision fatigue or cognitive overload.
- *Hyperstructures:* Multi-layered structures such as hypergraphs, representing complex interdependencies between options.

Example 4.206. Consider an online shopping platform like Amazon. Searching for *laptops* may yield thousands of options varying in brand, specifications, price, and user ratings. Each laptop has dependencies such as compatibility with accessories or warranties, creating a high-dimensional decision space.

Theorem 4.207. *Hyperdecision-making possesses the structure of a Hyperstructure.*

Proof. This follows directly from the definition. □

Theorem 4.208. *Hyperdecision-making generalizes classical decision-making by extending the choice space A, constraints C, and criteria K into hyperstructures, thereby accommodating multi-layered interdependencies and complex relationships.*

Proof. Classical decision-making operates over a finite set of alternatives A, constraints C, and criteria K, with decisions determined by:

$$a^* = \arg\max_{a \in A} \mathcal{U}(a, C, K),$$

where $\mathcal{U}$ is a utility function.

Hyperdecision-making generalizes this by replacing A, C, and K with hyperstructures:

$$\mathcal{H}(A) = \mathcal{P}_n^*(A), \quad \mathcal{H}(C), \quad \mathcal{H}(K),$$

and a utility function $\mathcal{U}_H : \mathcal{H}(A) \times \mathcal{H}(C) \times \mathcal{H}(K) \to \mathbb{R}$.

When $\mathcal{H}(A) = A$, $\mathcal{H}(C) = C$, and $\mathcal{H}(K) = K$, hyperdecision-making reduces to classical decision-making.

Therefore, hyperdecision-making generalizes classical decision-making by incorporating higher-dimensional structures and relationships. □

Definition 4.209 (Superhyperdecision-making)**.** Superhyperdecision-making is an extension of Hyperdecision-making, involving *multi-level hierarchical decision frameworks* where decisions span across recursive or nested (n-Superhyperstructures). It integrates:

- *n-Superhyperstructures:* Iterated, hierarchical decision spaces where each layer introduces new dimensions of complexity.
- *Context Adaptation:* Dynamic decision-making influenced by factors at multiple hierarchical levels.

Example 4.210 (n-SuperHyperDecision-Making in Global Climate Governance)**.** Global climate governance (cf. [552]) exemplifies a multi-level hierarchical decision-making process, where decisions at higher levels influence those at lower levels, as characteristic of n-SuperHyperDecision-Making:

- *Level* $n = 3$ *(Global Coordination):* International organizations, such as the United Nations Framework Convention on Climate Change (UNFCCC) [424], define global strategies, including:
  - Setting emission reduction targets.
  - Creating global carbon markets.
  - Negotiating international agreements.

  These decisions are shaped by:
  - Neutrosophic uncertainties about future climate projections.
  - Plithogenic constraints balancing economic disparities among nations.
  - The dynamic evolution of global consensus.
- *Level* $n = 2$ *(National Implementation):* National governments adapt global strategies to their specific contexts by implementing measures such as:
  - Carbon taxes.
  - Renewable energy subsidies.
  - Reforestation programs.

  This level involves:
  - Weighted criteria, such as cost-effectiveness, political feasibility, and public acceptance.
  - Interdependencies between measures (e.g., subsidies influencing carbon tax effectiveness).
  - Dynamic constraints, such as changing energy demands and resource availability.
- *Level* $n = 1$ *(Regional Execution):* Regional or local governments implement specific initiatives, including:
  - Constructing solar farms.
  - Introducing electric vehicle incentives.
  - Managing forest conservation projects.

  Decision-making at this level integrates:

  - Fuzzy criteria to handle factors like population density, resource accessibility, and local economic conditions.
  - Feedback adjustments based on community input or ecological monitoring.

- *Level $n = 0$ (Local Actions):* At the most granular level, individual or community-based efforts operationalize strategies, such as:
  - Installing solar panels on residential properties.
  - Organizing local climate education programs.
  - Conducting tree planting drives.

  These actions focus on optimizing immediate impact through practical and accessible implementations, guided by constraints such as budget limits and volunteer availability.

Dynamic Hierarchical Feedback: Information flows bidirectionally across all levels. For example:

- Evolving global scientific insights at $n = 3$ may necessitate policy adjustments cascading down to $n = 2$, $n = 1$, and $n = 0$.
- Feedback from local actions ($n = 0$) informs regional ($n = 1$) and national ($n = 2$) strategies, potentially influencing global priorities ($n = 3$).

This scenario exemplifies n-SuperHyperDecision-Making because:

- It spans four hierarchical levels ($n = 3$ to $n = 0$), each embedding decision complexity and interdependencies.
- Higher-level decisions impose constraints on lower levels, while lower-level feedback informs higher-level adaptations.
- The process integrates neutrosophic, plithogenic, and fuzzy approaches to manage uncertainties and support dynamic adaptation.

Achieving coherent and effective global climate governance requires integrating these multi-layered decisions into a unified n-SuperHyperstructure.

Theorem 4.211. *SuperHyperdecision-making inherently possesses the structure of a SuperHyperstructure.*

Proof. This result follows directly from the definition. □

Theorem 4.212. *Every instance of Superhyperdecision-making encapsulates at least one layer of Hyperdecision-making, but not all Hyperdecision-making scenarios generalize to Superhyperdecision-making.*

Proof.

1. *Base Case:* Hyperdecision-making operates within a hypergraph structure, where nodes represent choices, and hyperedges represent relationships or dependencies. This corresponds to $n = 1$-Superhyperstructures.
2. *Induction Hypothesis:* Assume that k-Superhyperdecision-making involves k-Superhyperstructures with hierarchical and recursive dependencies.
3. *Induction Step:* Adding another layer $k + 1$ introduces meta-decisions (e.g., weighting criteria dynamically), generalizing the decision space to $k + 1$-Superhyperstructures.

Thus, n-Superhyperdecision-making generalizes Hyperdecision-making for $n > 1$. □

Theorem 4.213 (Hierarchical Reduction in n-Superhyperdecision-making)**.** *In n-Superhyperdecision-making, achieving a final decision necessarily involves sequential reduction from the highest hierarchical level n to the base level n* $= 0$*, such that:*

$$\textit{Final Decision} = \bigcup_{k=0}^{n} \textit{Optimal Choices at Level } k.$$

Each reduction step $k \to k-1$ *resolves dependencies and constraints imposed by the higher levels, ensuring consistency across all levels.*

Proof. The theorem follows from the following steps:

1. *Base Case (*$n = 0$*):* At the base level, decisions are made within a single hyperstructure, involving only direct choices and constraints. No higher-level adjustments are necessary, as this is the foundational layer.
2. *Induction Hypothesis:* Assume that for a given $k = n$, decisions at this level depend on the results from $n + 1$, and resolving $n + 1$ ensures consistency across levels.
3. *Induction Step:* At $n = k + 1$, meta-decisions are made that introduce constraints and dependencies for $n = k$. By sequentially reducing from $k + 1$ to k, decisions at k are adapted based on results from $k + 1$, ensuring coherent propagation to $k - 1$.

Therefore, by induction, the final decision process necessarily involves reduction from n to 0, integrating and aligning decisions across all hierarchical levels. □

Example 4.214 (Multi-level Corporate Strategy (Superhyperdecision-making))**.** Corporate strategy involves defining an organization's long-term goals, resource allocation, and competitive positioning to maximize overall value and growth (cf. [160, 268, 445]). In a multinational corporation, the decision-making process spans multiple hierarchical levels:

- *At* $n = 4$ *(Global Strategy):* The global board establishes overarching strategic objectives, such as sustainability, profitability, and global market share. These objectives set the foundation for all subsequent decisions across the organization.
- *At* $n = 3$ *(Regional Alignment):* Regional divisions translate global goals into region-specific strategies. For example, they might prioritize market entry into emerging economies or adapt sustainability goals to regional regulations. Dependencies include aligning with global targets while accommodating regional constraints.
- *At* $n = 2$ *(Departmental Action Plans):* Departments within regional divisions, such as sales, R&D, and operations, develop detailed plans tailored to local markets. Examples include launching region-specific products, conducting localized marketing campaigns, or optimizing supply chain logistics.
- *At* $n = 1$ *(Team Execution):* Teams implement specific tasks, such as conducting customer outreach, deploying advertising campaigns, or rolling out new product lines. Execution at this level requires precise coordination to meet departmental goals and regional strategies while adhering to global objectives.
- *At* $n = 0$ *(Task-Level Implementation):* Individual contributors complete granular tasks essential for achieving team objectives. For instance, tasks might include designing marketing materials, coding software features, or negotiating supplier contracts. Decisions at this level are directly influenced by higher-level directives and are vital for overall coherence.

Hierarchical Dependency and Feedback: At each level, dependencies from higher levels must be resolved before progressing to the next. Feedback loops between levels ensure adaptability; for instance:

- Insights from $n = 0$ (e.g., task-level delays or resource shortages) inform adjustments at higher levels.

- Shifts in global priorities at $n = 4$ cascade down to influence decisions at all lower levels.

This hierarchical structure exemplifies n-Superhyperdecision-making by integrating global objectives with operational execution, ensuring consistency, adaptability, and alignment across all organizational levels.

Example 4.215 (Disaster Management (Superhyperdecision-making))**.** Disaster management involves planning, coordinating, and implementing strategies to mitigate, prepare for, respond to, and recover from natural or man-made disasters (cf. [113, 157, 381]). The decision-making process operates across multiple hierarchical levels:

- *At $n = 3$ (National Coordination):* The national government defines strategic priorities, such as allocating resources across regions, setting evacuation protocols, and establishing emergency communication systems. Decisions at this level are influenced by neutrosophic uncertainties, including incomplete forecasts and varying disaster severity across regions.
- *At $n = 2$ (Regional Prioritization):* Regional authorities adapt national strategies to local needs, prioritizing tasks such as evacuations, deployment of medical aid, or infrastructure repairs. This level incorporates weighted criteria, such as population density, vulnerability indices, and regional healthcare capacity.
- *At $n = 1$ (Local Execution):* Local teams carry out specific tasks, including setting up emergency shelters, distributing food and water, or coordinating search-and-rescue operations. Decisions at this level rely on fuzzy criteria, such as real-time resource availability and local feedback from affected communities.
- *At $n = 0$ (Task-Level Implementation):* Individual responders and volunteers perform granular tasks, such as transporting supplies, administering first aid, or setting up communication equipment. These actions directly support team-level objectives and ensure the timely execution of disaster response plans.

Hierarchical Integration and Feedback: At each level, higher-level priorities and constraints are integrated into actionable plans, while feedback loops ensure adaptability. For instance:

- Real-time updates from $n = 0$ (e.g., delays in supply delivery or changing weather conditions) inform adjustments at $n = 1$ and higher levels.
- Strategic shifts at $n = 3$ (e.g., reallocation of resources based on evolving disaster impact) cascade down to regional and local levels for implementation.

This multi-level framework illustrates n-Superhyperdecision-making, where coordinated actions at all hierarchical levels ensure effective disaster management, integrating strategic objectives with operational execution.

Example 4.216 (SuperHyperdecision-making on an Online Shopping Platform)**.** Consider an online shopping platform like Amazon (cf. [47, 237]). A user searching for *laptops* engages in a multi-layered decision-making process structured as follows:

- *Level $n = 0$ (Individual Choice):* The user evaluates a single product in detail, considering factors such as reviews, warranty terms, and compatibility with existing devices. This step represents the final choice among filtered options.
- *Level $n = 1$ (Filtered Decision):* Within the *Laptops* category, the user applies filters based on specifications such as brand, processor type, RAM size, and storage capacity. Weighted criteria, such as price versus performance, guide this filtering process.
- *Level $n = 2$ (Category Refinement):* The user narrows the search to a specific subcategory (e.g., gaming laptops or ultra-portables) based on broader considerations like intended use or budget constraints.
- *Level $n = 3$ (Category Selection):* At the highest level, the user selects the overarching category, such as *Laptops*, from a broader electronics section. Constraints at this level may include general preferences for type (e.g., laptops versus desktops) or purpose (e.g., gaming, work, or study).

Example 4.217 (Artificial Intelligence in Decision Support)**.** AI systems can model n-Superhyperdecision-making to assist in hierarchical decisions, such as autonomous multi-agent systems for supply chain optimization.

4.15 Discussions: Generalized n-Superhyperdecision-making

In this subsection, we explore the concept of Generalized n-Superhyperdecision-making based on the framework of the Generalized n-th Powerset. This approach establishes a foundation for decision-making processes where each decision step incorporates considerations such as Fuzzy, Neutrosophic, or Weighted criteria. Fuzzy Decision-Making [20,22,103,137,154,155,243,413,455,517,557] and Neutrosophic Decision-Making [117,192,394,566] are well-studied, making it a natural extension to examine the behavior of these concepts within the n-Superhyperdecision-making framework.

This framework aims to enhance the precision and adaptability of decision-making in complex, multi-dimensional environments. Future research is expected to focus on applying these concepts across various domains. Definitions and related concepts are provided below.

Definition 4.218 (Generalized n-Superhyperdecision-making)**.** Let D be a set of decision options, and let $G_n(D)$ denote the n-th generalized powerset of D as defined in [180]. A *Generalized n-Superhyperdecision-making framework* is defined as:

$$\mathcal{G}_n = (G_n(D), \diamond_n, C_n),$$

where:

- $G_n(D)$ represents the n-th level of generalized decisions, allowing subsets or enriched structures (e.g., fuzzy subsets, weighted subsets, neutrosophic subsets, plithogenic subsets).
- $\diamond_n : G_n(D) \times G_n(D) \to \mathcal{P}(G_n(D)) \setminus \{\emptyset\}$ is a hyperoperation that governs the interaction of decision elements, allowing multi-valued outcomes.
- C_n is a set of constraints or criteria that guide the decision process. These may include:
  - *Fuzzy constraints:* Membership degrees $\mu : D \to [0,1]$ assigned to decision options.
  - *Weighted criteria:* Priorities $w : D \to \mathbb{R}$ assigned to each option.
  - *Neutrosophic criteria:* Truth (T), indeterminacy (I), and falsity (F) values $T, I, F : D \to [0,1]$ satisfying $0 \le T(x) + I(x) + F(x) \le 3$.
  - *Plithogenic criteria:* For each decision $d \in D$, a set of attributes $\{v_i\}_{i=1}^m$ and their possible values $P(v_i)$. The decision compatibility is given by:

    $$pdf : D \times P(v) \to [0,1]^s, \quad pCF : P(v) \times P(v) \to [0,1]^t,$$

    where pdf is the degree of appurtenance, and pCF is the degree of contradiction between attributes.

The decision process involves sequential reductions from n to 0, with each step integrating the constraints C_k for $k = n, n-1, \ldots, 0$ to refine decision options and achieve an optimal decision set.

Example 4.219 (Generalized n-Superhyperdecision-making in Online Shopping)**.** Consider an online shopper browsing an e-commerce platform, such as Amazon, to purchase a laptop. The decision process involves multiple hierarchical levels, where each level integrates criteria like Fuzzy, Weighted, and Neutrosophic considerations.

- *Level* $n = 3$ *(General Preferences):* At this level, the shopper establishes broad preferences for the purchase:
  - *Fuzzy Criteria:* Assigning preference scores to brands:

    $$\mu(\text{Brand A}) = 0.8, \quad \mu(\text{Brand B}) = 0.6.$$

  - *Weighted Criteria:* Weighting key attributes:

    $$w(\text{Performance}) = 0.7, \quad w(\text{Price}) = 0.5, \quad w(\text{Portability}) = 0.6.$$

- *Neutrosophic Criteria:* Evaluating battery life with truth (T), indeterminacy (I), and falsity (F) values:
$$T = 0.9, \quad I = 0.1, \quad F = 0.0.$$

These preferences generate the decision set $G_3(D)$, containing laptops that align with the shopper's general preferences.

- *Level $n = 2$ (Feature-Specific Decisions):* The shopper refines their choices by focusing on specific features:
  - *Screen Size:* Prioritizing laptops with screen sizes between 13 and 15 inches:
  $$\mu(13\text{–}15 \text{ inches}) = 0.9.$$
  - *Processor:* Assigning weighted importance to processors:
  $$w(\text{Intel i7}) = 0.8, \quad w(\text{AMD Ryzen 7}) = 0.7.$$
  - *Compatibility:* Using Neutrosophic evaluation to assess software compatibility:
  $$T = 0.8, \quad I = 0.2, \quad F = 0.0.$$

  Applying these feature-specific constraints reduces $G_3(D)$ to $G_2(D)$, narrowing the decision space further.

- *Level $n = 1$ (Practical Constraints):* In the final step, practical constraints guide the shopper's choice:
  - *Budget:* The laptop must be priced between \$1000 and \$1500.
  - *Delivery Options:* Preference is given to laptops with fast delivery options.

  These constraints refine $G_2(D)$ into $G_1(D)$, containing the most suitable options or the single optimal choice.

- *Level $n = 0$ (Final Evaluation):* At this level, the shopper makes a detailed evaluation of the final choice, considering factors such as warranty terms, reviews, and compatibility with existing devices.

The process involves sequential reductions from $n = 3$ to $n = 0$, where each level integrates relevant criteria and constraints. This framework demonstrates the adaptability and precision of Generalized n-Superhyperdecision-making in complex, multi-dimensional decision scenarios.

Example 4.220 (Generalized n-Superhyperdecision-making in a National Healthcare System)**.** Consider the decision-making process in a national healthcare system (cf. [263, 520]):

- $n = 3$*: National Level Decisions* The central government prioritizes regions for resource allocation, such as vaccines or medical equipment, using:
  - *Fuzzy Criteria:* Regions with higher population density are assigned a higher priority:
  $$\mu(\text{Population Density}) = \text{High: } 0.8, \text{ Medium: } 0.5, \text{ Low: } 0.2.$$
  - *Weighted Metrics:* Budget constraints are factored in, assigning weights to regions based on projected costs:
  $$w(\text{Region A}) = 0.6, \quad w(\text{Region B}) = 0.4.$$
  - *Neutrosophic Uncertainty:* Future demand for healthcare services is evaluated using truth (T), indeterminacy (I), and falsity (F):
  $$T = 0.7, \quad I = 0.2, \quad F = 0.1.$$

  This level produces a prioritized list of regions for further refinement.

- $n = 2$*: State Level Decisions* State governments adapt national priorities to allocate resources among cities:
  - *Hospital Capacity:* Cities with higher available capacity are deprioritized to ensure equity.
  - *Critical Care Needs:* Weighting critical care requirements:

$$w(\text{City X}) = 0.7, \quad w(\text{City Y}) = 0.5.$$

  For example, if Region A prioritizes Cities X and Y, the state's resource allocation ensures alignment with national goals.

- $n = 1$*: Facility Level Decisions* Local facilities make operational decisions:
  - *Procurement:* Ordering medical supplies based on immediate demand.
  - *Staff Scheduling:* Adjusting work shifts for peak periods of care.

  These tasks directly implement strategies from higher levels.

- $n = 0$*: Individual Decision-Making* At this level, individual healthcare workers make on-the-ground decisions:
  - *Patient Triage:* Deciding the order of patient care based on severity.
  - *Resource Allocation:* Allocating limited equipment like ventilators or ICU beds to patients.

  These decisions are influenced by immediate circumstances and local constraints while adhering to guidelines from higher levels.

The decision-making process involves sequential reductions from $n = 3$ to $n = 0$. Each step integrates constraints C_k from higher levels, ensuring a consistent and efficient allocation of resources across all levels.

Example 4.221 (Generalized *n*-Superhyperdecision-making in Product Development)**.** Consider the decision-making process in product development (cf. [82, 195, 308]):

- $n = 3$*: Strategic Decisions* The executive team determines broad product categories using plithogenic criteria:
  - *Customer Preferences:* Attributes such as durability and aesthetics are analyzed with compatibility evaluations:
$$\text{Compatibility}(Durability, Aesthetics) = 0.85.$$
  - *Technological Trends:* Products are ranked based on projected market adoption rates and innovation indices.

- $n = 2$*: Tactical Decisions* The product development team evaluates design prototypes:
  - *Cost-Performance Trade-offs:* Weighted criteria are applied to balance production costs with expected performance:

$$w(\text{Prototype 1}) = 0.8, \quad w(\text{Prototype 2}) = 0.6.$$

  - *Reliability Metrics:* Prototypes are tested under various conditions to evaluate robustness.

  The selected prototype aligns with strategic goals and available resources.

- $n = 1$*: Operational Decisions* Manufacturing decisions are made to optimize the production process:
  - *Material Sourcing:* Suppliers are chosen based on cost, quality, and reliability.
  - *Production Scheduling:* Assembly lines are scheduled to meet delivery deadlines while minimizing downtime.

- $n = 0$*: Individual Decisions* Workers on the production line make on-the-spot decisions during manufacturing:

– *Quality Control:* Identifying and addressing defects in real-time to ensure product standards.
– *Equipment Handling:* Adjusting machinery settings based on specific batch requirements.

These decisions are guided by operational protocols and immediate situational factors.

By integrating plithogenic criteria at each level, this process ensures comprehensive evaluations, accommodating multi-attribute complexities across all hierarchical levels, down to individual actions.

Example 4.222 (Superhyperdecision-making in Emergency Pandemic Response)**.** Emergency pandemic response [349, 553] requires a multi-level hierarchical decision-making framework, where each level is characterized by distinct objectives, constraints, and interactions:

- *Level* $n = 3$ *(Global Coordination):* International organizations such as the World Health Organization (WHO) define global strategies, including vaccine distribution priorities, international travel restrictions, and research funding for variant surveillance. Constraints at this level often include *neutrosophic uncertainties* due to incomplete or contradictory data about emerging virus variants, their transmissibility, and vaccine efficacy.
- *Level* $n = 2$ *(National Response):* National governments allocate resources such as vaccines, ventilators, and testing kits to regional authorities. *Plithogenic constraints* play a significant role, balancing factors like economic impact (e.g., GDP reduction), public health outcomes (e.g., infection fatality rates), and political feasibility (e.g., public trust or opposition).
- *Level* $n = 1$ *(Regional Implementation):* Regional health authorities focus on prioritizing specific interventions, such as expanding ICU capacities, establishing mass vaccination centers, and deploying mobile testing units. Decisions are guided by *fuzzy criteria* that account for population density, infection rates, and regional healthcare capacity.
- *Level* $n = 0$ *(Local Action):* Local teams execute concrete tasks, such as administering vaccines at community health centers, delivering medical supplies, or conducting public awareness campaigns. These actions are optimized using *weighted constraints*, such as minimizing logistical costs while maximizing vaccination coverage and accessibility.

By systematically reducing complexity from $n = 3$ to $n = 0$, the decision-making process integrates global strategies into practical, actionable local plans. This hierarchical approach ensures adaptability and consistency while addressing challenges at each level.

Example 4.223 (Superhyperdecision-making in Urban Planning)**.** Urban planning (cf. [93, 219, 220]) involves a multi-level hierarchical decision-making framework that incorporates both global objectives and local execution:

- *Level* $n = 4$ *(National Level):* The national government establishes high-level urban development goals, such as increasing affordable housing, promoting renewable energy adoption, and reducing traffic congestion. These goals often involve *plithogenic attributes*, requiring trade-offs between cost efficiency, environmental impact, and political support.
- *Level* $n = 3$ *(Regional Level):* Regional authorities adapt national goals to local contexts by setting specific targets, such as allocating budgets for solar energy projects or defining housing quotas for major metropolitan areas. *Fuzzy constraints* arise from uncertainties in factors like population growth, economic trends, and regional resource availability.
- *Level* $n = 2$ *(City Level):* City councils develop detailed action plans, including selecting contractors, revising zoning regulations, and planning public infrastructure improvements. Decisions at this level are driven by *weighted criteria*, such as maximizing cost-effectiveness, ensuring social equity, and minimizing environmental disruption.
- *Level* $n = 1$ *(Neighborhood Level):* Local teams carry out tangible tasks, such as constructing residential complexes, installing solar panels on public buildings, or developing bike lanes. Decisions at this level are often influenced by *neutrosophic uncertainties*, such as potential delays, fluctuating material costs, or changing community preferences.

- *Level $n = 0$ (Community Action):* Community organizations and individual stakeholders implement hyperlocal initiatives, such as organizing community clean-ups, planting trees, or participating in neighborhood watch programs. Actions at this level are shaped by *specific constraints and goals*, like enhancing local livability, fostering civic engagement, and addressing immediate neighborhood concerns.

Each level systematically incorporates constraints and dependencies from higher levels, allowing for a consistent and adaptable decision-making framework. This ensures that national objectives are effectively translated into impactful local initiatives and hyperlocal actions.

Theorem 4.224. *In a Generalized n-Superhyperdecision-making process, the optimal decision set D^* is obtained through sequential reductions:*

$$D^* = \bigcap_{k=0}^{n} R_k(D, C_k),$$

where R_k is the refinement operation at level k, integrating the constraints C_k.

Proof. We proceed by induction on the levels k.

At the highest level n, the decision set D is reduced using the refinement operation R_n, which integrates the constraints C_n. This results in a subset $R_n(D, C_n) \subseteq D$, containing only those decisions that satisfy the constraints at level n.

Assume that for level $k+1$, the refinement operation R_{k+1} has reduced the decision set to $R_{k+1}(D, C_{k+1})$, satisfying all constraints at levels $k+1, k+2, \ldots, n$. At level k, the refinement operation R_k further reduces this subset by integrating the constraints C_k, resulting in:

$$R_k(R_{k+1}(D, C_{k+1}), C_k).$$

Since R_k ensures that the constraints C_k are satisfied, the resulting decision set satisfies all constraints from level k to level n.

At the base level $k = 0$, the refinement operation R_0 ensures that the constraints C_0 are satisfied, reducing the decision set to:

$$R_0(R_1(D, C_1), C_0).$$

This ensures that all constraints from levels 0 to n are integrated.

By recursively applying the refinement operations R_k for all $k = n, n-1, \ldots, 0$, the final decision set is given by:

$$D^* = \bigcap_{k=0}^{n} R_k(D, C_k),$$

ensuring that all constraints from every level k are satisfied. This process guarantees the optimal decision set D^*, completing the proof. □

Theorem 4.225. *Generalized n-Superhyperdecision-making extends classical decision-making by incorporating multi-level hyperoperations and enriched constraints, making it adaptable to complex, multi-attribute decision scenarios.*

Proof. Classical decision-making operates on a single decision set D with a utility function and a set of constraints. Generalized n-Superhyperdecision-making expands this by introducing:

- *Hierarchical Decision Sets:* The decision space is extended to $G_n(D)$, where each level represents increasingly complex subsets or enriched structures (e.g., fuzzy subsets, neutrosophic subsets).
- *Multi-level Constraints:* Constraints C_k at each level k allow for the incorporation of advanced criteria, such as fuzzy membership degrees, weights, or plithogenic attributes.

- *Hyperoperations:* The operation $\diamond_n$ facilitates multi-valued relationships and interactions between decisions, capturing dependencies and complexities not possible in classical frameworks.

By combining these elements, the framework can handle scenarios with multi-level decision dependencies, conflicting criteria, and uncertainty. This adaptability demonstrates its extension beyond classical decision-making frameworks. □

Theorem 4.226. *Generalized n-SuperHyperdecision-making inherently possesses the structure of a SuperHyperstructure.*

Proof. From the definition of Generalized n-Superhyperdecision-making, the decision framework involves:

- $G_n(D)$, the n-th generalized powerset of the decision set D, forming a hierarchical decision space.
- $\diamond_n$, a multi-level hyperoperation acting on $G_n(D)$, enabling recursive and multi-valued interactions.
- C_n, constraints defined across n levels, incorporating enriched structures such as fuzzy, neutrosophic, and plithogenic attributes.

These elements align with the definition of a SuperHyperstructure, where the components $G_n(D)$, $\diamond_n$, and C_n represent hierarchical levels, hyperoperations, and associated constraints, respectively. Therefore, Generalized n-SuperHyperdecision-making naturally possesses the structure of a SuperHyperstructure. □

Funding

This study did not receive any external financial support.

Acknowledgments

We extend our heartfelt gratitude to all individuals whose guidance and encouragement contributed to the successful completion of this research. We are also deeply thankful to our readers for their interest in this work and to the authors of the cited references, whose foundational contributions have greatly enriched our study.

Data Availability

As this study is purely theoretical and mathematical, no data analysis was conducted. We encourage future researchers to explore related empirical analyses or data-driven investigations as needed.

Ethical Approval

This research focuses entirely on theoretical and mathematical aspects, involving no experiments with human participants or animals.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this study.

Disclaimer

This study presents theoretical advancements that have not yet been practically tested or applied. We encourage future researchers to validate and refine these methods through empirical research. While we have made every effort to ensure accuracy and proper citation, unintentional errors or omissions may still occur. Readers are advised to independently verify the referenced materials. The interpretations and views expressed in this paper are solely those of the authors and do not necessarily represent the views of their affiliated institutions.

References

[1] Mujahid Abbas, Ghulam Murtaza, and Florentin Smarandache. *Basic operations on hypersoft sets and hypersoft point*. Infinite Study, 2020.

[2] Pelumi Abimbola, Anirudh Sanga, and Shweta Mongia. Hadoop framework ecosystem: Ant solution to an elephantic data. *SSRN Electronic Journal*, 2019.

[3] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark T. Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *HUC*, 1999.

[4] Milton Abramowitz, Irene A. Stegun, and David Miller. Handbook of mathematical functions with formulas, graphs and mathematical tables (national bureau of standards applied mathematics series no. 55). *Journal of Applied Mechanics*, 32:239–239, 1965.

[5] Alex Abreu and Antonio Nigro. A symmetric function of increasing forests. *Forum of Mathematics, Sigma*, 9, 2020.

[6] Joshua Ackerman and George V. Cybenko. A survey of neural networks and formal languages. *ArXiv*, abs/2006.01338, 2020.

[7] Anu Adhikari. Full stack javascript: Web application development with mean. 2016.

[8] William A. Adkins and Steven H. Weintraub. Algebra: An approach via module theory. 1992.

[9] Fakhry Asad Agusfrianto, Saba Al-Kaseasbeh, Mariam Hariri, and Yudi Mahatma. On neutrohyperrings and neutroorderedhyperrings. *Neutrosophic Sets and Systems*, 77:1–19, 2025.

[10] Fakhry Asad Agusfrianto, Madeleine Al-Tahan, Mariam Hariri, and Yudi Mahatma. Examples of neutrohyperstructures on biological inheritance. *Neutrosophic Sets and Systems*, 60:583–592, 2023.

[11] Fakhry Asad Agusfrianto, Madeleine Al Tahan, and Yudi Mahatma. An introduction to neutrohyperstructures on some chemical reactions. In *NeutroGeometry, NeutroAlgebra, and SuperHyperAlgebra in Today's World*, pages 81–96. IGI Global, 2023.

[12] David Ayinde Akinpelu, Oluwaseun Mujidat Adekoya, Peter Olusakin Oladoye, Chukwuma C. Ogbaga, and Jude A. Okolie. Machine learning applications in biomass pyrolysis: from biorefinery to end-of-life product management. *Digital Chemical Engineering*, 2023.

[13] Sinan G Aksoy, Cliff Joslyn, Carlos Ortiz Marrero, Brenda Praggastis, and Emilie Purvine. Hypernetwork science via high-order hypergraph walks. *EPJ Data Science*, 9(1):16, 2020.

[14] Salah Hasan Saleh Al-subhi, Iliana Pérez Pupo, Roberto García Vacacela, Pedro Piñero, Pérez, and Maikel Yelandi Leyva Vázquez. A new neutrosophic cognitive map with neutrosophic sets on connections, application in project management. *Neutrosophic Sets and Systems*, 22:63–75, 2018.

[15] M Al Tahan and B Davvaz. Hv-module of functions over hv-ring of arithmetics and it's fundamental module. *Analele Stiintifice ale Universitatii Ovidius Constanta: Seria Matematica*, 32(1), 2024.

[16] M Al Tahan and Bijan Davvaz. Weak chemical hyperstructures associated to electrochemical cells. *Iranian Journal of Mathematical Chemistry*, 9(1):65–75, 2018.

[17] Madeleine Al Tahan, Saba Al-Kaseasbeh, and Bijan Davvaz. Neutrosophic quadruple hv-modules and their fundamental module. *Neutrosophic Sets and Systems*, 72:304–325, 2024.

[18] Madeleine Al-Tahan and Bijan Davvaz. Chemical hyperstructures for elements with four oxidation states. *Iranian Journal of Mathematical Chemistry*, 13(2):85–97, 2022.

[19] Madeleine Al-Tahan, Bijan Davvaz, Florentin Smarandache, and Osman Anis. On some neutrohyperstructures. *Symmetry*, 13(4):535, 2021.

[20] Fatemeh Alamroshan, Mahyar La'li, and Mohsen Yahyaei. The green-agile supplier selection problem for the medical devices: a hybrid fuzzy decision-making approach. *Environmental Science and Pollution Research International*, 29:6793 – 6811, 2021.

[21] Giovanni S. Alberti and Luigi Ambrosio. A geometrical approach to monotone functions in r n. 2007.

[22] Ghaliah Alhamzi, Saman Javaid, Umer Shuaib, Abdul Razaq, Harish Garg, and Asima Razzaque. Enhancing interval-valued pythagorean fuzzy decision-making through dombi-based aggregation operators. *Symmetry*, 15:765, 2023.

[23] K.R. Alieva. Modern banking and strategic portfolio management. *Herald of Dagestan State University*, 2022.

[24] José Aliste-Prieto, Anna de Mier, Rosa C. Orellana, and José Zamora. Marked graphs and the chromatic symmetric function. 2022.

[25] Michael S. Allen. Business portfolio management: Valuation, risk assessment, and eva strategies. 1999.

[26] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. Madgraph 5: going beyond. *Journal of High Energy Physics*, 2011:1–40, 2011.

[27] Alimzhan Amanov and Damir Yeliussizov. Macmahon's statistics on higher-dimensional partitions. *Forum of Mathematics, Sigma*, 11, 2020.

[28] Reza Ameri, M Amiri-Bideshki, Sarka Hoskova-Mayerova, and AB Saeid. Distributive and dual distributive elements in hyperlattices. 2017.

[29] Reza Ameri, M Amiri-Bideshki, Arsham Borumand Saeid, and Sarka Hoskova-Mayerova. Prime filters of hyperlattices. 2016.

[30] Reza Ameri and Morteza Norouzi. On commutative hyperrings. *International journal of algebraic hyperstructures and its applications*, 1(1):45–58, 2014.

[31] Omid Amini, Frédéric Mazoit, Nicolas Nisse, and Stéphan Thomassé. Submodular partition functions. *Discrete Mathematics*, 309(20):6000–6008, 2009.

[32] Setyaningsih Ana and P.Elav arasi. Topology. *International Journal of Mathematics Trends and Technology*, 2018.

[33] Thomas Andersson, Mikael Cäker, Stefan Tengblad, and Mikael Wickelgren. Building traits for organizational resilience through balancing organizational structures. *Scandinavian Journal of Management*, 35(1):36–45, 2019.

[34] Katie Angliss and Pete Harpum. Strategic portfolio management. 2022.

[35] SM Anvariyeh and B Davvaz. Strongly transitive geometric spaces associated with (m, n)-ary hypermodules. *Scientific Annals of" Al. L. Cuza" University of Iasi*, 2013.

[36] Ioannis D. Apostolopoulos and Peter P. Groumpos. Fuzzy cognitive maps: Their role in explainable artificial intelligence. *Applied Sciences*, 2023.

[37] Michael Armbrust, Reynold Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei A. Zaharia. Spark sql: Relational data processing in spark. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015.

[38] Kenneth J. Arrow. Social choice and individual values. 1951.

[39] Charles Ashbacher. *Introduction to Neutrosophic logic*. Infinite Study, 2014.

[40] Michael Francis Atiyah and Ian G. MacDonald. Introduction to commutative algebra. 1969.

[41] Roger Atkinson. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management*, 17:337–342, 1999.

[42] Arbaz Attar, Pranay Mule, Piyush Kulkarni, Shubham Narale, and Prof. Ms. Jaitee Bankar. Investment portfolio management system: A survey. *International Journal for Research in Applied Science and Engineering Technology*, 2023.

[43] Maurice Auslander and Mark Bridger. Stable module theory. 1969.

[44] James Ax. The elementary theory of finite fields. *Annals of Mathematics*, 88:239, 1968.

[45] Robert Axelrod. Structure of decision : the cognitive maps of political elites. 2015.

[46] Rosa M. Badia, Javier Conejero, Carlos Diaz, Jorge Ejarque, Daniele Lezzi, Francesc Lordan, Cristian Ramon-Cortes, and Raül Sirvent. Comp superscalar, an interoperable programming framework. *SoftwareX*, 3:32–36, 2015.

[47] H Manjula Bai. Perception analysis of online shopping: A case study of amazon.com. 2020.

[48] Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018.

[49] Sudipto Banerjee, Bradley P. Carlin, and Alan E. Gelfand. Hierarchical modeling and analysis for spatial data. 2003.

[50] Tamer Basar and Geert Jan Olsder. Dynamic noncooperative game theory. 1982.

[51] C. W. J. Beenakker. Random-matrix theory of quantum transport. *Reviews of Modern Physics*, 69:731–808, 1996.

[52] Z Belali, SM Anvariyeh, and S Mirvakili. Free and cyclic canonical $\{(m,n)-\}$ ary hypermodules. *Tamkang Journal of Mathematics*, 42(1):105–118, 2011.

[53] Mariana Belgiu and Lucian Daniel Dragut. Random forest in remote sensing: A review of applications and future directions. *Isprs Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[54] Richard Bellman. Introduction to matrix analysis. 1972.

[55] Martin P. Bendsøe and Ole Sigmund. Topology optimization: ”theory, methods, and applications”. 2011.

[56] Peter Bennett. Hypergames: Developing a model of conflict. *Futures*, 12:489–507, 1980.

[57] Peter M. Bennett and Malcolm Dando. Complex strategic analysis: A hypergame study of the fall of france. *Journal of the Operational Research Society*, 30:23–32, 1979.

[58] Nathanael Berestycki, Eyal Lubetzky, Yuval Peres, and Allan Sly. Random walks on the random graph. *arXiv: Probability*, 2015.

[59] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[60] Niclas Bergman. *Recursive Bayesian estimation: Navigation and tracking applications*. PhD thesis, Linköping University, 1999.

[61] Abraham Berman and Robert J. Plemmons. Nonnegative matrices in the mathematical sciences. In *Classics in Applied Mathematics*, 1979.

[62] Ute Beyer. Algebra an approach via module theory. 2016.

[63] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. *Oper. Res.*, 69:950–973, 2018.

[64] Gérard Biau and Erwan Scornet. A random forest guided tour. *TEST*, 25:197 – 227, 2015.

[65] John W Bidewell. Decision making in personal investment. 2007.

[66] Jeff A. Bilmes. Submodularity in machine learning and artificial intelligence. *ArXiv*, abs/2202.00132, 2022.

[67] Garrett Birkhoff. *Lattice theory*, volume 25. American Mathematical Soc., 1940.

[68] Ethan D Bloch. *Proofs and fundamentals: a first course in abstract mathematics*. Springer Science & Business Media, 2011.

[69] Evan Bolyen, Jai Ram Rideout, Matthew R. Dillon, Nicholas A. Bokulich, Christian C. Abnet, Gabriel A. Al-Ghalith, Harriet Alexander, Eric J. Alm, Manimozhiyan Arumugam, Francesco Asnicar, Yang Bai, Jordan E. Bisanz, Kyle Bittinger, Asker Daniel Brejnrod, Colin J. Brislawn, C. Titus Brown, Benjamin J. Callahan, Andres Mauricio Caraballo-Rodriguez, John H. Chase, Emily K. Cope, Ricardo R. da Silva, Christian Diener, Pieter C. Dorrestein, Gavin M. Douglas, Daniel M. Durall, Claire Duvallet, Christian F. Edwardson, Madeleine Ernst, Mehrbod Estaki, Jennifer T. Fouquier, Julia M. Gauglitz, Sean M. Gibbons, Deanna L. Gibson, Antonio Gonzalez, Kestrel Gorlick, Jiarong Guo, Benjamin M Hillmann, Susan P. Holmes, Hannes Holste, Curtis Huttenhower, Gavin A. Huttley, Stefan Janssen, Alan K. Jarmusch, Lingjing Jiang, Benjamin D. Kaehler, Kyo Bin Kang, Christopher R. Keefe, Paul S Keim, Scott T. Kelley, Dan Knights, Irina Koester, Tomasz Kosciolek, Jorden Kreps, Morgan G. I. Langille, Joslynn S. Lee, Ruth E. Ley, Yong xin Liu, Erikka Loftfield, Catherine A. Lozupone, Massoud Maher, Clarisse A. Marotz, Bryan D. Martin, Daniel McDonald, Lauren J. McIver, Alexey V. Melnik, Jessica L. Metcalf, Sydney C. Morgan, James T. Morton, Ahmad Turan Naimey, Jose A Navas-Molina, Louis-Felix Nothias, Stephanie B. Orchanian, Talima R. Pearson, Samuel L. Peoples, Daniel Petras, Mary Lai Preuss, Elmar Pruesse, Lasse Buur Rasmussen, Adam R. Rivers, Michael Scott Robeson, Patrick Rosenthal, N. Segata, Michael Shaffer, Arron Shiffer, Rashmi Sinha, Se Jin Song, John R. Spear, Austin D. Swafford, Luke R. Thompson, Pedro J. Torres, Pauline Trinh, Anupriya Tripathi, Peter J. Turnbaugh, Sabah Ul-Hasan, Justin J. J. van der Hooft, Fernando Vargas, Y. Vazquez-Baeza, Emily Vogtmann, Max von Hippel, William Walters, Yunhu Wan, Mingxun Wang, Jonathan Warren, Kyle C. Weber, Charles Hall Davis Williamson, Amy D. Willis, Zhenjiang Zech Xu, Jesse R. Zaneveld, Yilong Zhang, Qiyun Zhu, Rob Knight, and J. Gregory Caporaso. Reproducible, interactive, scalable and extensible microbiome data science using qiime 2. *Nature Biotechnology*, 37:852 – 857, 2019.

[70] Borzoo Bonakdarpour and Sarai Sheinvald. Automata for hyperlanguages. *arXiv preprint arXiv:2002.09877*, 2020.

[71] Borzoo Bonakdarpour and Sarai Sheinvald. Finite-word hyperlanguages. *Information and Computation*, 295:104944, 2023.

[72] Ferdaous Bouaziz and Naveed Yaqoob. Rough hyperfilters in po-la-semihypergroups. *Discrete Dynamics in Nature and Society*, 2019:1–8, 2019.

[73] Barry Bozeman. Handbook of interview research: Context and method. *Evaluation and Program Planning*, 26:37–39, 2003.

[74] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[75] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.

[76] Walter Briec and Qi Bin Liang. On some semilattice structures for production technologies. *European Journal of Operational Research*, 215(3):740–749, 2011.

[77] William A. Buz Brock. Discrete choice with social interactions. *IEEE Transactions on Information Theory*, 2001.

[78] William A. Buz Brock, Cars H. Hommes, and J. Barkley Rosser. A rational route to randomness. 1997.

[79] Thomas A. Brody, Jorge Flores, J. B. French, Pier A. Mello, Akhilesh Pandey, and S. S. M. Wong. Random matrix physics: Spectrum and strength fluctuations. *Reviews of Modern Physics*, 53:385–479, 1981.

[80] Frederick P. Brooks, Albert L. Hopkins, Peter G. Neumann, and William V. Wright. An experiment in musical composition. *IRE Trans. Electron. Comput.*, 6:175–182, 1957.

[81] Shona L Brown and Kathleen M Eisenhardt. Product development: Past research, present findings, and future directions. *Academy of management review*, 20(2):343–378, 1995.

[82] Shona L. Brown and Kathleen M. Eisenhardt. Product development: Past research, present findings, and future directions. *Academy of Management Review*, 20:343–378, 1995.

[83] Sid Browne. Reaching goals by a deadline: digital options and continuous-time active portfolio management. *Advances in Applied Probability*, 31:551 – 577, 1996.

[84] Sid Browne. Risk-constrained dynamic active portfolio management. *Management Science*, 46:1188–1199, 2000.

[85] Richard Hubert Bruck. *A survey of binary systems*, volume 20. Springer, 1971.

[86] Dietrich Burde. Group theory. *Computers, Rigidity, and Moduli*, 2019.

[87] M Burgin. Hyperfunctionals and generalized distributions. In *Stochastic Processes and Functional Analysis*, pages 109–132. CRC Press, 2004.

[88] Rod M Burstall and John Darlington. A transformation system for developing recursive programs. *Journal of the ACM (JACM)*, 24(1):44–67, 1977.

[89] Tony Bush. Theories of educational leadership and management. 2020.

[90] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[91] Colin Camerer. Behavioral game theory: Experiments in strategic interaction. 2003.

[92] Colin Camerer. Individual decision making. *The Handbook of Experimental Economics*, 2020.

[93] Scott Campbell. Planning: Green cities, growing cities, just cities? urban planning and the contradictions of sustainable development. *The Earthscan Reader in Sustainable Cities*, 2021.

[94] Joao Paulo Carvalho. On the semantics and the use of fuzzy cognitive maps and dynamic cognitive maps in social sciences. *Fuzzy Sets Syst.*, 214:6–19, 2013.

[95] Y. V. M. Cepeda, M. A. R. Guevara, E. J. J. Mogro, and R. P. Tizano. Impact of irrigation water technification on seven directories of the san juan-patoa river using plithogenic n-superhypergraphs based on environmental indicators in the canton of pujili, 2021. *Neutrosophic Sets and Systems*, 74:46–56, 2024.

[96] Ivan Chajda, Radomir Halas, and Jan Kuhr. *Semilattice structures*, volume 30. Heldermann Lemgo, 2007.

[97] Prasanna Chandra. Investment analysis and portfolio management. 2004.

[98] Kiran Kumar Chandriah and Raghavendra V. Naraganahalli. Maximizing a deep submodular function optimization with a weighted max-sat problem for trajectory clustering and motion segmentation. *Applied Intelligence*, 51:8192 – 8211, 2021.

[99] Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *Artificial Intelligence Review*, 57(9):250, 2024.

[100] Chandra Chekuri, Jan Vondr'ak, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011.

[101] Long Chen. Finding structure with randomness : Probabilistic algorithms for constructing. 2016.

[102] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374, 2021.

[103] Shyi-Ming Chen and Jiann-Mean Tan. Handling multicriteria fuzzy decision-making problems based on vague set theory. *Fuzzy sets and systems*, 67(2):163–172, 1994.

[104] John Child. Organizational structure, environment and performance: The role of strategic choice. *sociology*, 6(1):1–22, 1972.

[105] Hyeonyoung Choi, Eunju Ko, Eun Young Kim, and Pekka Mattila. The role of fashion brand authenticity in product management: : a holistic marketing approach. *Journal of Product Innovation Management*, 32:233–242, 2015.

[106] Andrzej Chrzeszczyk and Jan Kochanowski. Matrix computations. In *Encyclopedia of Parallel Computing*, 2011.

[107] Gruia Clinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40:1740–1766, 2011.

[108] Grigore Clugreanu. Lattice concepts of module theory. 2000.

[109] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. Portfolio management in new product development: Lessons from the leaders-i. *Research-technology Management*, 40:16–28, 1997.

[110] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. New product portfolio management : Practices and performance. 1999.

[111] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. New problems, new solutions: Making portfolio management more effective. *Research-Technology Management*, 43:18 – 33, 2000.

[112] Robert G. Cooper, Scott J. Edgett, and Elko J. Kleinschmidt. Portfolio management for new product development: results of an industry practices study. *R & D Management*, 31:361–380, 2001.

[113] Damon P. Coppola. Introduction to international disaster management. 2006.

[114] Piergiulio Corsini and Violeta Leoreanu. *Applications of hyperstructure theory*, volume 5. Springer Science & Business Media, 2013.

[115] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon Asher Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[116] Merle Crawford and Anthony Di Benedetto. *New products management*. McGraw-Hill, 2015.

[117] Ridvan cSahin. Multi-criteria neutrosophic decision making method based on score and accuracy functions under neutrosophic environment. 2014.

[118] Cameron Davidson-Pilon. lifelines: survival analysis in python. *J. Open Source Softw.*, 4:1317, 2019.

[119] B Davvaz. Weak algebraic hyperstructures as a model for interpretation of chemical reactions. *Iranian Journal of Mathematical Chemistry*, 7(2):267–283, 2016.

[120] B Davvaz, A Dehghan Nezhad, and A Benvidi. Chemical hyperalgebra: Dismutation reactions. *Match-Communications in Mathematical and Computer Chemistry*, 67(1):55, 2012.

[121] Bijan Davvaz, Wieslaw A Dudek, and Thomas Vougiouklis. A generalization of n-ary algebraic systems. *Communications in Algebra*, 37(4):1248–1263, 2009.

[122] Bijan Davvaz and Narjes Firouzkouhi. Fundamental relation on fuzzy hypermodules. *Soft Computing*, 23:13025 – 13033, 2019.

[123] Bijan Davvaz, A Dehghan Nezhad, and SM Moosavi Nejad. Algebraic hyperstructure of observable elementary particles including the higgs boson. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 90(1):169–176, 2020.

[124] Bijan Davvaz and Thomas Vougiouklis. *Walk Through Weak Hyperstructures, A: Hv-structures*. World Scientific, 2018.

[125] David Vaughan Day. Leadership development:: A review in context. *Leadership Quarterly*, 11:581–613, 2000.

[126] José R González de Mendívil. A generalization of myhill–nerode theorem for fuzzy languages. *Fuzzy Sets and Systems*, 301:103–115, 2016.

[127] Somen Debnath. *Introduction to Hyper Neutrosophic Set*. Infinite Study, 2020.

[128] OR Dehghan. Linear functionals on hypervector spaces. *Filomat*, 34(9):3031–3043, 2020.

[129] OR Dehghan. An introduction to neutrohypervector spaces. *Neutrosophic Sets and Systems*, 58(1):21, 2023.

[130] Aurore Delaigle and Peter Hall. Defining probability density for a distribution of random functions. *arXiv: Statistics Theory*, 2010.

[131] Gregory G Dess and Donald W Beard. Dimensions of organizational task environments. *Administrative science quarterly*, pages 52–73, 1984.

[132] Levi DeValve, Sasa Pekec, and Yehua Dennis Wei. Approximate submodularity in network design problems. *OPER: Continuous (Topic)*, 2021.

[133] Giuseppe Di Maio and Somashekhar Naimpally. Comparison of hypertopologies. 1991.

[134] Reinhard Diestel. Graduate texts in mathematics: Graph theory.

[135] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173(33):12, 2005.

[136] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[137] Hasan Dincer, Serhat Yuksel, and Serkan Eti. Identifying the right policies for increasing the efficiency of the renewable energy transition with a novel fuzzy decision-making model. *Journal of Soft Computing and Decision Analytics*, 2023.

[138] A Dramalidis, N Lygeros, and T Vougiouklis. On the iso-hyper-representation theory.

[139] Achilles Dramalidis and Thomas Vougiouklis. Fuzzy hv-substructures in a two dimensional euclidean vector space. *Iranian Journal of Fuzzy Systems*, 6(4):1–9, 2009.

[140] Ralf Dresner. Rethinking context language as an interactive phenomenon. 2016.

[141] Martin A. Dufwenberg. Game theory. *Wiley interdisciplinary reviews. Cognitive science*, 2 2:167–173, 2011.

[142] William R. Duncan. A guide to the project management body of knowledge. 1996.

[143] Richard Durrett. Random graph dynamics: References. 2006.

[144] Frank H Easterbrook and Daniel R Fischel. *The economic structure of corporate law*. Harvard university press, 1996.

[145] Imad eddine Meredef, Mohamed Yacine Hammoudi, Abir Betka, Madina Hamiane, and Khalida Mimoune. Stability and stabilization of ts fuzzy systems via line integral lyapunov fuzzy function. *Electronics*, 2022.

[146] Ward Edwards. The theory of decision making. *Psychological bulletin*, 514:380–417, 1954.

[147] Philip Ehrlich. *Real numbers, generalizations of the reals, and theories of continua*, volume 242. Springer Science & Business Media, 2013.

[148] David Eisenbud. Commutative algebra: with a view toward algebraic geometry. 1995.

[149] Bonnie Eisenman. Learning react native: Building native mobile apps with javascript. 2016.

[150] Adam J. Engler, Adam J. Engler, Shamik Sen, Shamik Sen, H. Lee Sweeney, and Dennis E. Discher. Matrix elasticity directs stem cell lineage specification. *Cell*, 126:677–689, 2006.

[151] Paul Erdos. Asymmetric graphs.

[152] Ali Akbar Estaji and Fereshteh Bayati. On rough sets and hyperlattices. *Ratio Mathematica*, 34:15, 2018.

[153] Guillaume Evanno, Sebastien Regnaut, and Jérôme Goudet. Detecting the number of clusters of individuals using the software structure: a simulation study. *Molecular ecology*, 14(8):2611–2620, 2005.

[154] Payam Safaei Fakhri, Omid Asghari, Sliva Sarspy, Mehran Borhani Marand, Paria Moshaver, and Mohammad Trik. A fuzzy decision-making system for video tracking with multiple objects in non-stationary conditions. *Heliyon*, 9, 2023.

[155] Alireza Fallahpour, Sina Nayeri, Mohammad Sheikhalishahi, K. Wong, Guangdong Tian, and Amir Mohammad Fathollahi-Fard. A hyper-hybrid fuzzy decision-making framework for the sustainable-resilient supplier selection problem: a case study of malaysian palm oil industry. *Environmental Science and Pollution Research International*, pages 1 – 21, 2021.

[156] Benson Farb, Mark Kisin, and Jesse Wolfson. Modular functions and resolvent problems. *Mathematische Annalen*, 386:113–150, 2019.

[157] Bill Faulkner. Towards a framework for tourism disaster management. *Tourism Management*, 22:135–147, 2001.

[158] Michael C. Ferris, Steven P. Dirkse, Jan-Hendrick Jagla, and Alexander Meeraus. An extended mathematical programming framework. *Comput. Chem. Eng.*, 33:1973–1982, 2009.

[159] Wallace Feurzeig and Seymour Papert. Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19:487 – 501, 2011.

[160] Charles J. Fombrun and Mark T. Shanley. What's in a name? reputation building and corporate strategy. *Academy of Management Journal*, 33:233–258, 1990.

[161] Jeff Forcier, Paul Bissex, and Wesley J. Chun. Python web development with django. 2008.

[162] Guy Fowler. Multiplicative independence of modular functions. *Journal de Théorie des Nombres de Bordeaux*, 2020.

[163] Jeffrey A Frankel. Regional trading blocs in the world economic system. *Institute for International Economics*, 1997.

[164] Luigi Fratta and Ugo Montanari. A boolean algebra method for computing the terminal reliability in a communication network. *IEEE Transactions on Circuit Theory*, 20:203–211, 1973.

[165] Hadar Frenkel and Sarai Sheinvald. Realizable and context-free hyperlanguages. *arXiv preprint arXiv:2209.10306*, 2022.

[166] Samuel Fricker. Software product management. In *Requirements Engineering: Foundation for Software Quality*, 2017.

[167] Stefan Friedl. Algebraic topology. *Graduate Studies in Mathematics*, 2020.

[168] Takaaki Fujita. Breaking down barriers: Proposals for overcoming challenges in student project management. *European Journal of Management and Marketing Studies*, 2023.

[169] Takaaki Fujita. A consideration of project-based learning: Exploring its potential, effects, challenges, and best practices. *European Journal of Education Studies*, 10(3), 2023.

[170] Takaaki Fujita. Reconsideration and proposal of development models in projects-"quasi" development models: Quasi-waterfall and quasi-agile. *European Journal of Social Sciences Studies*, 9(2), 2023.

[171] Takaaki Fujita. Expanding horizons of plithogenic superhyperstructures: Applications in decision-making, control, and neuro systems. 2024.

[172] Takaaki Fujita. Reconsideration of neutrosophic social science and neutrosophic phenomenology with non-classical logic. 2024.

[173] Takaaki Fujita. Review of some superhypergraph classes: Directed, bidirected, soft, and rough. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[174] Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.

[175] Takaaki Fujita. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. *arXiv preprint arXiv:2412.01176*, 2024.

[176] Takaaki Fujita. Theoretical interpretations of large uncertain and hyper language models: Advancing natural uncertain and hyper language processing. 2024.

[177] Takaaki Fujita. Various properties of various ultrafilters, various graph width parameters, and various connectivity systems. *arXiv preprint arXiv:2408.02299*, 2024.

[178] Takaaki Fujita. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*. Biblio Publishing, 2025.

[179] Takaaki Fujita. A comprehensive discussion on fuzzy hypersoft expert, superhypersoft, and indetermsoft graphs. *Neutrosophic Sets and Systems*, 77:241–263, 2025.

[180] Takaaki Fujita. Exploration of graph classes and concepts for superhypergraphs and n-th power mathematical structures. 2025.

[181] Takaaki Fujita and Smarandache Florentin. Some graph parameters for superhypertree-width and neutrosophictree-width. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Third Volume)*. Biblio Publishing, 2024.

[182] Takaaki Fujita and Florentin Smarandache. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[183] Takaaki Fujita and Florentin Smarandache. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Third Volume)*. Biblio Publishing, 2024.

[184] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

[185] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[186] Takaaki Fujita and Florentin Smarandache. Introduction to upside-down logic: Its deep relation to neutrosophic logic and applications. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Third Volume)*. Biblio Publishing, 2024.

[187] Takaaki Fujita and Florentin Smarandache. A review of the hierarchy of plithogenic, neutrosophic, and fuzzy graphs: Survey and applications. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[188] Takaaki Fujita and Florentin Smarandache. A short note for hypersoft rough graphs. *HyperSoft Set Methods in Engineering*, 3:1–25, 2024.

[189] Takaaki Fujita and Florentin Smarandache. Uncertain automata and uncertain graph grammar. *Neutrosophic Sets and Systems*, 74:128–191, 2024.

[190] M Ganesh. *Introduction to fuzzy sets and fuzzy logic*. PHI Learning Pvt. Ltd., 2006.

[191] Junyu Gao, Mengyuan Chen, Liangyu Xiang, and Changsheng Xu. A comprehensive survey on evidential deep learning and its applications. *arXiv preprint arXiv:2409.04720*, 2024.

[192] Harish Garg and Nancy. Algorithms for possibility linguistic single-valued neutrosophic decision-making based on copras and aggregation operators with new information measures. *Measurement*, 2019.

[193] Jim Geelen, Bert Gerards, and Geoff Whittle. Obstructions to branch-decomposition of matroids. *Journal of Combinatorial Theory, Series B*, 96(4):560–570, 2006.

[194] David Gems. The hyperfunction theory: an emerging paradigm for the biology of aging. *Ageing research reviews*, 74:101557, 2022.

[195] Donald Gerwin and Nicholas J. Barrowman. An evaluation of research on integrated product development. *Manag. Sci.*, 48:938–953, 2002.

[196] Patrizia Ghisellini, Catia Cialani, and Sergio Ulgiati. A review on circular economy: the expected transition to a balanced interplay of environmental and economic systems. *Journal of Cleaner production*, 114:11–32, 2016.

[197] M. Ghods, Zahra Rostami, and Florentin Smarandache. Introduction to neutrosophic restricted superhypergraphs and neutrosophic restricted superhypertrees and several of their properties. 2022.

[198] Masoud Ghods, Zahra Rostami, and Florentin Smarandache. Introduction to neutrosophic restricted superhypergraphs and neutrosophic restricted superhypertrees and several of their properties. *Neutrosophic Sets and Systems*, 50:480–487, 2022.

[199] Jayanta Ghosh and Tapas Kumar Samanta. Hyperfuzzy sets and hyperfuzzy group. *Int. J. Adv. Sci. Technol*, 41:27–37, 2012.

[200] María Ángeles Gil. Fuzzy random variables. *Inf. Sci.*, 133:1–2, 2001.

[201] William Nelson Goetzmann, Richard C. Grinold, and Ronald N. Kahn. Active portfolio management. 1999.

[202] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. In *JACM*, 1986.

[203] Gene H. Golub. Matrix computations. 1983.

[204] Igor I. Gorban. Randomness and hyper-randomness. 2017.

[205] Igor I. Gorban. Determinism, uncertainty, randomness, and hyper-randomness. 2018.

[206] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 21–32, 1999.

[207] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: A survey. In *Mathematical Foundations of Computer Science 2001: 26th International Symposium, MFCS 2001 Mariánské Lázne, Czech Republic, August 27–31, 2001 Proceedings 26*, pages 37–57. Springer, 2001.

[208] Fernando Q. Gouvea. A guide to groups, rings, and fields: Groups and their representations. 2012.

[209] George Gratzer. Lattice theory: First concepts and distributive lattices. 1971.

[210] George Gratzer. *General lattice theory*. Springer Science & Business Media, 2002.

[211] Darij Grinberg. An introduction to graph theory. 2013.

[212] Richard C. Grinold and Ronald N. Kahn. Active portfolio management : a quantitative approach for providing superior returns and controlling risk. 2000.

[213] Jonathan L Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.

[214] Anil K Gupta and Vijay Govindarajan. Business unit strategy, managerial characteristics, and business unit effectiveness at strategy implementation. *Academy of Management journal*, 27(1):25–41, 1984.

[215] Isabelle M Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[216] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[217] Irmeli Halinen. The new educational curriculum in finland. *Improving the quality of childhood in Europe*, 7:75–89, 2018.

[218] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53:217–288, 2009.

[219] P. Hall and Mark Tewdwr-Jones. Urban and regional planning. 2019.

[220] Peter Hall. Cities of tomorrow: An intellectual history of urban planning and design since 1880. 2014.

[221] Paul Halmos and Steven Givant. *Introduction to Boolean algebras*. Springer, 2009.

[222] Paul R Halmos. *Lectures on Boolean algebras*. Courier Dover Publications, 2018.

[223] Paul Richard Halmos. *Naive set theory*. van Nostrand, 1960.

[224] Mohammad Hamidi and Marzieh Rahmati. On binary decision hypertree (hyperdiagram). *AUT Journal of Mathematics and Computing*, 5(2):117–130, 2024.

[225] Mohammad Hamidi, Florentin Smarandache, and Elham Davneshvar. Spectrum of superhypergraphs via flows. *Journal of Mathematics*, 2022(1):9158912, 2022.

[226] Mohammad Hamidi and Mohadeseh Taghinezhad. *Application of Superhypergraphs-Based Domination Number in Real World*. Infinite Study, 2023.

[227] Michael T Hannan and John Freeman. Structural inertia and organizational change. *American sociological review*, pages 149–164, 1984.

[228] Ronald M Harden, Susette Sowden, and William R Dunn. Educational strategies in curriculum development: the spices model. *Medical education*, 18(4):284–297, 1984.

[229] Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'io, Marcy Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585:357 – 362, 2020.

[230] Theodore Edward Harris. Counting measures, monotone random set functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 10:102–119, 1968.

[231] William E. Hart, Jean-Paul Watson, and David L. Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3:219–260, 2011.

[232] James K Harter, Frank L Schmidt, and Theodore L Hayes. Business-unit-level relationship between employee satisfaction, employee engagement, and business outcomes: a meta-analysis. *Journal of applied psychology*, 87(2):268, 2002.

[233] Pengfei He and Xiaolong Xin. Fuzzy hyperlattices. *Computers & Mathematics with Applications*, 62(12):4682–4690, 2011.

[234] Pengfei He, Xiaolong Xin, and Jianming Zhan. On rough hyperideals in hyperlattices. *Journal of Applied Mathematics*, 2013(1):915217, 2013.

[235] R. Hema, R. Sudharani, and M. Kavitha. A novel approach on plithogenic interval valued neutrosophic hypersoft sets and its application in decision making. *Indian Journal Of Science And Technology*, 2023.

[236] E. Hendukolaii. On fuzzy homomorphisms between hypernear-rings. 2011.

[237] Yan Heng, Zhifeng Gao, Yuan Jiang, and Xuqi Chen. Exploring hidden factors behind online food shopping from amazon reviews: A topic mining approach. *Journal of Retailing and Consumer Services*, 42:161–168, 2018.

[238] Kate Hevner. The affective value of pitch and tempo in music. *The American Journal of Psychology*, 49(4):621–630, 1937.

[239] Kostaq Hila, Serkan Onar, Bayram Ali Ersoy, and Bijan Davvaz. On generalized intuitionistic fuzzy subhyperalgebras of boolean hyperalgebras. *Journal of Inequalities and Applications*, 2013:1–15, 2013.

[240] Peter G Hinman. *Recursion-theoretic hierarchies*, volume 9. Cambridge University Press, 2017.

[241] Crawford S Holling. Understanding the complexity of economic, ecological, and social systems. *Ecosystems*, 4:390–405, 2001.

[242] Doeko Homan. Very basic set theory. 2023.

[243] Dug Hun Hong and Chang-Hwan Choi. Multicriteria fuzzy decision-making problems based on vague set theory. *Fuzzy sets and systems*, 114(1):103–113, 2000.

[244] John E. Hopcroft and Jeffrey D. Ullman. Introduction to automata theory, languages and computation. 1979.

[245] Roger A. Horn and Charles R. Johnson. Matrix analysis. 1985.

[246] Torsten Hothorn and Achim Zeileis. partykit: A modular toolkit for recursive partytioning in r. *The Journal of Machine Learning Research*, 16(1):3905–3909, 2015.

[247] Zhe Hou. Automata theory and formal languages. *Texts in Computer Science*, 2021.

[248] James Thomas House and George V. Cybenko. Hypergame theory applied to cyber attack and defense. In *Defense + Commercial Sensing*, 2010.

[249] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[250] Lorenz Huelsbergen. Learning recursive sequences via evolution of machine-language programs. *Genetic Programming*, pages 186–194, 1997.

[251] JE Humphreys. On the hyperalgebra of a semisimple algebraic group. In *Contributions to Algebra*, pages 203–210. Elsevier, 1977.

[252] John Y. Hung, Wei bing Gao, and James C. Hung. Variable structure control: a survey. *IEEE Trans. Ind. Electron.*, 40:2–22, 1993.

[253] Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926, 2018.

[254] Isao Imai. *Applied hyperfunction theory*, volume 8. Springer Science & Business Media, 2013.

[255] Ozan Irsoy and Claire Cardie. Deep recursive neural networks for compositionality in language. *Advances in neural information processing systems*, 27, 2014.

[256] Nikolay Ivanov, Tatyana Fedoseeva, and Olga Mezentseva. Transformation of a small company's organizational structure in the construction industry. In *Building Life-cycle Management. Information Systems and Technologies: Selected Papers*, pages 373–385. Springer, 2022.

[257] Kenneth E Iverson. A programming language. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, pages 345–351, 1962.

[258] Rishabh K. Iyer and Jeff A. Bilmes. Submodular point processes with applications to machine learning. In *International Conference on Artificial Intelligence and Statistics*, 2015.

[259] Elina Jaakkola and Anette Hallin. Organizational structures for new service development. *Journal of Product Innovation Management*, 35(2):280–297, 2018.

[260] Muhammad Naveed Jafar, Muhammad Haris Saeed, Kainat Muniba Khan, Faten S. Alamri, and Hamiden A. Wahed Khalifa. Distance and similarity measures using max-min operators of neutrosophic hypersoft sets with application in site selection for solid waste management systems. *IEEE Access*, PP:1–1, 2022.

[261] Sirus Jahanpanah and Roohallah Daneshpayeh. On derived superhyper be-algebras. *Neutrosophic Sets and Systems*, 57(1):21, 2023.

[262] Sirus Jahanpanah and Roohallah Daneshpayeh. An outspread on valued logic superhyperalgebras. *Facta Universitatis, Series: Mathematics and Informatics*, pages 427–437, 2024.

[263] Miroslav Jankj and Jan Voracek. Dynamic modelling of national healthcare system. *Measuring Business Excellence*, 19:76–89, 2015.

[264] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[265] Yanjun Jiang, Yijing Wang, Ruiqi Yang, and Weina Ye. Maximizing approximately non-k-submodular monotone set function with matroid constraint. In *Theory and Applications of Models of Computation*, 2022.

[266] Zhong-Ping Jiang and Henk Nijmeijer. A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Transactions on Automatic control*, 44(2):265–279, 1999.

[267] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. *ArXiv*, abs/1907.05388, 2019.

[268] Gerry Johnson, Kevan Scholes, and Richard Whittington. Exploring corporate strategy. 1988.

[269] Thomas Johnsson. *Lambda lifting: Transforming programs to recursive equations*. Springer, 1985.

[270] Calvin Jongsma. Basic set theory and combinatorics. *Undergraduate Texts in Mathematics*, 2019.

[271] Jaiung Jun. Geometry of hyperfields. *Journal of Algebra*, 2017.

[272] Jaiung Jun. Algebraic geometry over hyperrings. *Advances in Mathematics*, 323:142–192, 2018.

[273] Li Jun. Continuity of monotone set function and convergences of measurable function sequence. *Fuzzy systems and mathematics*, 2005.

[274] Young Bae Jun, Kul Hur, and Kyoung Ja Lee. Hyperfuzzy subalgebras of bck/bci-algebras. *Annals of Fuzzy Mathematics and Informatics*, 2017.

[275] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. *ArXiv*, abs/1809.02745, 2018.

[276] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pages 3183–3191. PMLR, 2019.

[277] Shizuo Kamo. Non-standard real number systems with regular gaps. *Tsukuba journal of mathematics*, 5:21–24, 1981.

[278] P Kamporoudi and T Vougiouklis. Cyclicity in some classes of hv-groups. *Journal of Algebraic Hyperstructures and Logical Algebras*, 1(3):91–97, 2020.

[279] Dr.W.B.Vasantha Kandasamy and Florentin Smarandache. Fuzzy cognitive maps and neutrosophic cognitive maps. 2003.

[280] T Kaplani and T Vougiouklis. On strong-inverse elements. *Journal of Algebraic Hyperstructures and Logical Algebras*, 1(3):51–60, 2020.

[281] Abdullah Kargın, Florentin Smarandache, and Memet Şahin. *New Type Hyper Groups, New Type SuperHyper Groups and Neutro-New Type SuperHyper Groups*. Infinite Study, 2023.

[282] Gregory Karpilovsky. Topics in field theory. 1989.

[283] M. Kaur and A. Singh. Approach to multi-criteria decision-making in a neutrosophic picture hyper-soft set environment using generalized neutrosophic topsis. *Neutrosophic Sets and Systems*, 67(1):5, 2024.

[284] Michael Keith. Random forest. *Machine Learning with Regression in Python*, 2021.

[285] Gerald I. Kendall and Steven C. Rollins. Advanced project portfolio management and the pmo: Multiplying roi at warp speed. 2003.

[286] Marc Kery and J. Andrew Royle. Applied hierarchical modeling in ecology: Analysis of distribution, abundance and species richness in r and bugs. 2015.

[287] Harold Kerzner. Project management workbook to accompany project management : a systems approach to planning, scheduling and controlling. 2001.

[288] Mohamed Khaled. Topics in algebra. *AMS/MAA Textbooks*, 2021.

[289] H. E. Khalid, G. D. Gungor, and M. A. N. Zainal. Neutrosophic superhyper bi-topological spaces: Original notions and new insights. *Neutrosophic Sets and Systems*, 51:33–45, 2022.

[290] N. U. Khan, Saddam Husain, and Owais Khan. A novel kind of beta logarithmic function and their properties. *Hacettepe Journal of Mathematics and Statistics*, 2022.

[291] Sabahat A. Khan, Mohammad Y. Abbasi, Kostaq Hila, and Aakif F. Talee. A study of soft hyperideals in right regular la-semihypergroups. *Asian-European Journal of Mathematics*, 2023.

[292] Waheed Ahmad Khan, Waqar Arif, Hossein Rashmanlou, and Saeed Kosari. Interval-valued picture fuzzy hypergraphs with application towards decision making. *Journal of Applied Mathematics and Computing*, 70(2):1103–1125, 2024.

[293] Mounir M. El Khatib, Asma Al Mulla, and Wadha Al Ketbi. The role of blockchain in e-governance and decision-making in project and program management. *Advances in Internet of Things*, 2022.

[294] B. Melvin Kiernan. The development of galois theory from lagrange to artin. *Archive for History of Exact Sciences*, 8:40–154, 1971.

[295] Stephen Cole Kleene. General recursive functions of natural numbers. *Mathematische annalen*, 112(1):727–742, 1936.

[296] Israel Kleiner. Modern algebra and the rise of mathematical structures. *Historia Mathematica*, 28:304–312, 2001.

[297] Erich Peter Klement, Madan Lal Puri, and Dan A. Ralescu. Limit theorems for fuzzy random variables. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 407:171 – 182, 1986.

[298] Alexander Kock and Hans Georg Gemünden. Antecedents to decision-making quality and agility in innovation portfolio management. *Journal of Product Innovation Management*, 33:670–686, 2016.

[299] BBN Koguep and C Lele. On hyperlattices: congruence relations, ideals and homomorphism. *Afrika Matematika*, 30(1):101–111, 2019.

[300] Robert L. Kohl. An introduction to algebraic topology. 2016.

[301] Manuel Kolp, Paolo Giorgini, and John Mylopoulos. Multi-agent architectures as organizational structures. *Autonomous Agents and Multi-Agent Systems*, 13:3–25, 2006.

[302] Bart Kosko. Fuzzy cognitive maps. *Int. J. Man Mach. Stud.*, 24:65–75, 1986.

[303] Nicholas Kovach, Alan S. Gibson, and Gary B. Lamont. Hypergame theory: A model for conflict, misperception, and deception. 2015.

[304] Claire J. Kramsch. Context and culture in language teaching. 1993.

[305] Marc Krasner. A class of hyperrings and hyperfields. *International Journal of Mathematics and Mathematical Sciences*, 6:307–311, 1983.

[306] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability*, 2014.

[307] Jochen Krebs. Agile portfolio management. 2008.

[308] Vish Krishnan and Karl T. Ulrich. Product development decisions: A review of the literature. *Management Science*, 47:1–21, 2001.

[309] Carol L Krumhansl. Rhythm and pitch in music cognition. *Psychological bulletin*, 126(1):159, 2000.

[310] Maxim V. Kuleshov, Matthew R. Jones, Andrew D. Rouillard, Nicolas F. Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L. Jenkins, Kathleen M. Jagodnik, Alexander Lachmann, Michael G. McDermott, Caroline D. Monteiro, Gregory W. Gundersen, and Avi Ma'ayan. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Research*, 44:W90 – W97, 2016.

[311] Joachim Lambek. Lectures on rings and modules. 1976.

[312] M Lathamaheswari, S Sudha, Said Broumi, and Florentin Smarandache. New aspects of neutrosuperhyper algebra with its application. In *NeutroGeometry, NeutroAlgebra, and SuperHyperAlgebra in Today's World*, pages 17–51. IGI Global, 2023.

[313] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. 1998.

[314] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

[315] Daniel D. Lee, H. Sebastian, and Seung y. Algorithms for non-negative matrix factorization. In *Neural Information Processing Systems*, 2000.

[316] Miljan Lekovic. Choice between active and passive portfolio management strategy. *Ekonomski izazovi*, 2022.

[317] Johannes Lengler. A general dichotomy of evolutionary algorithms on monotone functions. *IEEE Transactions on Evolutionary Computation*, 24:995–1009, 2018.

[318] Robert W. Lent, Steven D. Brown, and Gail Hackett. Contextual supports and barriers to career choice: A social cognitive analysis. *Journal of Counseling Psychology*, 47:36–49, 2000.

[319] Maikel Leon. Aggregating procedure for fuzzy cognitive maps. *The International FLAIRS Conference Proceedings*, 2023.

[320] Wassily W Leontief. Quantitative input and output relations in the economic systems of the united states. *The review of economic statistics*, pages 105–125, 1936.

[321] Paul Blain Levy. Wiener's random function, and other laplacian random functions. 1951.

[322] Jun Yu Li. Order continuous of monotone set function and convergence of measurable functions sequence. *Appl. Math. Comput.*, 135:211–218, 2003.

[323] James R Lincoln, Mitsuyo Hanada, and Kerry McBride. Organizational structures in japanese and us manufacturing. *Administrative Science Quarterly*, pages 338–364, 1986.

[324] Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1491–1500, 2014.

[325] Yian-Kui Liu and Baoding Liu. Fuzzy random variables: A scalar expected value operator. *Fuzzy Optimization and Decision Making*, 2:143–160, 2003.

[326] Martin Loebl. Introduction to graph theory. 2010.

[327] John Lombardi. The department/division structure in the community college. 1973.

[328] Falko Lorenz and Silvio V. F. Levy. Fields with structure, algebras and advanced topics. 2008.

[329] Heila Lotz-Sisitka, John Fien, and Mphemelang Ketlhoilwe. Traditions and new niches: An overview of environmental education curriculum and learning research. *International handbook of research on environmental education*, pages 194–205, 2013.

[330] Gilles Louppe, Kyunghyun Cho, Cyril Becot, and Kyle Cranmer. Qcd-aware recursive neural networks for jet physics. *Journal of High Energy Physics*, 2019(1):1–23, 2019.

[331] Robert Lowen and Mark Sioen. Proximal hypertopologies revisited. *Set-Valued Analysis*, 6:1–19, 1998.

[332] Michael Luby and Charles Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). *SIAM J. Comput.*, 17:373–386, 1986.

[333] Roberto Lucchetti. Hypertopologies and applications. *Recent Developments in Well-Posed Variational Problems*, pages 193–209, 1995.

[334] Justina Lydekaityte and Torben Tambo. Smart packaging: definitions, models and packaging as an intermediator between digital and physical product management. *The International Review of Retail, Distribution and Consumer Research*, 30:377 – 410, 2020.

[335] Veronica E. Lynn, Niranjan Balasubramanian, and H. A. Schwartz. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[336] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ArXiv*, abs/1611.00712, 2016.

[337] Pradip Kumar Maji, Ranjit Biswas, and A Ranjan Roy. Soft set theory. *Computers & mathematics with applications*, 45(4-5):555–562, 2003.

[338] Pinaki Majumdar. Neutrosophic sets and its applications to decision making. 2015.

[339] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374, 2016.

[340] Thomas W Malone and Stephen A Smith. Modeling the performance of organizational structures. *Operations Research*, 36(3):421–436, 1988.

[341] James G. March and Chip Heath. A primer on decision making : how decisions happen. 1994.

[342] Benedetto Marelli, Nereus Patel, Thomas Duggan, Giovanni Perotto, Elijah Shirman, Chunmei Li, David L Kaplan, and Fiorenzo G Omenetto. Programming function into mechanical forms by directed assembly of silk bulk materials. *Proceedings of the National Academy of Sciences*, 114(3):451–456, 2017.

[343] Miia Martinsuo and Päivi Hoverfält. Change program management: Toward a capability for managing value-oriented, integrated multi-project change in its context. *International Journal of Project Management*, 36:134–146, 2018.

[344] F. Marty. Sur une generalisation de la notion de groupe. In *Proceedings of the 8th Congress Math. Scandinaves*, pages 45–49, Stockholm, Sweden, 1934.

[345] Christos Massouros and Gerasimos Massouros. An overview of the foundations of the hypergroup theory. 2021.

[346] Gerasimos Massouros and Christos Massouros. State machines and hypergroups. *Mathematics*, 2022.

[347] James Maxwell and Ben Smith. Geometry of tropical extensions of hyperfields. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 2023.

[348] John McCarthy. Recursive functions of symbolic expressions and their computation by machine, part i. *Communications of the ACM*, 3(4):184–195, 1960.

[349] Jodie McVernon, Tania C. Sorrell, Jenny Firman, Brendan Murphy, and Sharon R. Lewin. Is australia prepared for the next pandemic? *Medical Journal of Australia*, 206, 2017.

[350] Jack R. Meredith and Samuel J. Mantel. Project management: A managerial approach. 1989.

[351] Neri Merhav and Igal Sason. An integral representation of the logarithmic function with applications in information theory. *Entropy*, 22, 2019.

[352] Yuan Miao, Zhi qiang Liu, Shi Li, and Chee Kheong Siew. Dynamical cognitive network-an extension of fuzzy cognitive map. *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pages 43–46, 1999.

[353] M. Mikelionyte and A. Lezgovko. Gender impact on personal investment strategies. *Economics and Culture*, 18:32 – 45, 2021.

[354] Charles F. Miller. Combinatorial group theory. 2002.

[355] Irwin Miller. Probability, random variables, and stochastic processes. *Technometrics*, 8:378–380, 1966.

[356] Henry Mintzberg. The design school: reconsidering the basic premises of strategic management. *Strategic management journal*, 11(3):171–195, 1990.

[357] E. J. Mogro, J. R. Molina, G. J. S. Canas, and P. H. Soria. Tree tobacco extract (*Nicotiana glauca*) as a plithogenic bioinsecticide alternative for controlling fruit fly (*Drosophila immigrans*) using n-superhypergraphs. *Neutrosophic Sets and Systems*, 74:57–65, 2024.

[358] Jesper M. Møller. General topology. 2008.

[359] Dmitriy Molodtsov. Soft set theory-first results. *Computers & mathematics with applications*, 37(4-5):19–31, 1999.

[360] Kalyan Mondal, Surapati Pramanik, and Nandalal Ghosh. A study on problems of hijras in west bengal based on neutrosophic cognitive maps. 2014.

[361] Mark H Moore. *Creating public value: Strategic management in government*. Harvard university press, 1997.

[362] Mitsuo Morimoto. *An introduction to Sato's hyperfunctions*, volume 129. American Mathematical Soc., 1993.

[363] Maedeh Motameni, Reza Ameri, and R. Sadeghi. Hypermatrix based on krasner hypervector spaces. 2013.

[364] Ghulam Muhiuddin, E. Mohammadzadeh, Fahimeh Mohammadzadeh, and R. A. Borzooei. Construction of fuzzy hyper bci-algebras using fuzzy hypergroupoid. *New Math. Nat. Comput.*, 19:557–577, 2022.

[365] David E. Muller. Application of boolean algebra to switching circuit design and to error detection. *Trans. I R E Prof. Group Electron. Comput.*, 3:6–12, 1954.

[366] Yuya Murakami. A continuity of cycle integrals of modular functions. *The Ramanujan Journal*, 55:1177–1187, 2019.

[367] Sagvan Y Musa and Baravan A Asaad. Topological structures via bipolar hypersoft sets. *Journal of Mathematics*, 2022(1):2896053, 2022.

[368] Roger B. Myerson. Game theory - analysis of conflict. 1991.

[369] Hiroshi Nagamochi, Takashi Shiraki, and Toshihide Ibaraki. Augmenting a submodular and posi-modular set function by a multigraph. *Journal of Combinatorial Optimization*, 5:175–212, 2001.

[370] Somashekhar Naimpally. All hypertopologies are hit-and-miss. *Applied General Topology*, 3(1):45–53, 2002.

[371] James B. Nation. Notes on lattice theory. 1998.

[372] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14:265–294, 1978.

[373] Ariel Neufeld and Philipp Schmocker. Universal approximation property of banach space-valued random feature models including random neural networks. 2023.

[374] Anthony Newcomb. 6. action and agency in mahler's ninth symphony, second movement. 2019.

[375] Luan Nguyen-Huynh and Tai Vo-Van. A new fuzzy time series forecasting model based on clustering technique and normal fuzzy function. *Knowledge and Information Systems*, 65:3489–3509, 2023.

[376] P Nikolaidou. Hyperstructures on bar of v & v in pieces. *Journal of Algebraic Hyperstructures and Logical Algebras*, 1(3):73–79, 2020.

[377] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. Algorithmic game theory. 2007.

[378] Morteza Norouzi and Reza Ameri. Some new directions in soft (fuzzy) hypermodules. *Fuzzy Information and Engineering*, 14:167 – 181, 2022.

[379] Volodymyr Novykov, Christopher Bilson, Adrian Gepp, Geoff Harris, and Bruce James Vanstone. Deep learning applications in investment portfolio management: a systematic literature review. *Journal of Accounting Literature*, 2023.

[380] Jeannie Oakes. *Keeping track: How schools structure inequality*. Yale University Press, 2005.

[381] Geoff OBrien, Phil Okeefe, Joanne Rose, and Ben Wisner. Climate change and disaster management. *Disasters*, 301:64–80, 2015.

[382] Omalsad Hamood Odhah, Huda M. Alshanbari, Zubair Ahmad, Faridoon Khan, and Abdal-Aziz H. El-Bagoury. A novel probabilistic approach based on trigonometric function: Model, theory with practical applications. *Symmetry*, 15:1528, 2023.

[383] Gulay Oguz and Bijan Davvaz. Soft topological hyperstructure. *J. Intell. Fuzzy Syst.*, 40:8755–8764, 2021.

[384] Travis E. Oliphant. Guide to numpy. 2015.

[385] S. Onar. A note on neutrosophic soft set over hyperalgebras. *Symmetry*, 16(10):1288, 2024.

[386] Izhar Oppenheim. Topological methods in group theory. 2018.

[387] James B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118:237–251, 2007.

[388] Rodolfo González Ortega, Marcos David Oviedo Rodríguez, Maikel Yelandi Leyva Vázquez, Jesús Estupiñán, Ricardo, João Alcione Sganderla Figueiredo, and Florentin Smarandache. Pestel analysis based on neutrosophic cognitive maps and neutrosophic numbers for the sinos river basin management. 2019.

[389] Martin J. Osborne and Ariel Rubinstein. A course in game theory. 1995.

[390] E. Ostrom. A behavioral approach to the rational choice theory of collective action: Presidential address, american political science association, 1997. *American Political Science Review*, 92:1 – 22, 1998.

[391] Elinor Ostrom. Beyond markets and states: polycentric governance of complex economic systems. *American economic review*, 100(3):641–672, 2010.

[392] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[393] Sang-il Oum and Paul Seymour. Certifying large branch-width. In *Symposium on Discrete Algorithms: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, volume 22, pages 810–813. Citeseer, 2006.

[394] Dragan Pamucar, Morteza Yazdani, Radojko Obradović, Anil Kumar, and Mercedes Torres. A novel fuzzy hybrid neutrosophic decision-making approach for the resilient supplier selection problem. *International Journal of Intelligent Systems*, 35:1934 – 1986, 2020.

[395] Elpiniki I. Papageorgiou. Review study on fuzzy cognitive maps and their applications during the last decade. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 828–835, 2011.

[396] Elpiniki I. Papageorgiou and Jose L. Salmeron. A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, 21:66–79, 2013.

[397] Omiros Papaspiliopoulos. High-dimensional probability: An introduction with applications in data science. *Quantitative Finance*, 20:1591 – 1594, 2020.

[398] Athanasios Papoulis. Probability, random variables and stochastic processes. 1965.

[399] Athanasios Papoulis and S. Unnikrishna Pillai. Probability, random variables, and stochastic processes. 2002.

[400] Anil Singh Parihar, Om Prakash Verma, and Chintan Khanna. Fuzzy-contextual contrast enhancement. *IEEE Transactions on Image Processing*, 26:1810–1819, 2017.

[401] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703, 2019.

[402] Peerasit Patanakul. How to achieve effectiveness in project portfolio management. *IEEE Transactions on Engineering Management*, 69:987–999, 2022.

[403] Peter Paule and Cristian-Silviu Radu. Holonomic relations for modular functions and forms: First guess, then prove. *International Journal of Number Theory*, 2020.

[404] Swaraj Pawar. Foodies full stack web application: Connecting you to your college canteens. *International Journal for Research in Applied Science and Engineering Technology*, 2024.

[405] Zdzis law Pawlak. Rough sets. *International journal of computer & information sciences*, 11:341–356, 1982.

[406] Zdzislaw Pawlak. Rough set theory and its applications to data analysis. *Cybernetics & Systems*, 29(7):661–688, 1998.

[407] Zdzis law Pawlak. Rough sets and intelligent data analysis. *Information sciences*, 147(1-4):1–12, 2002.

[408] Zdzis law Pawlak. *Rough sets: Theoretical aspects of reasoning about data*, volume 9. Springer Science & Business Media, 2012.

[409] Zdzislaw Pawlak, Jerzy Grzymala-Busse, Roman Slowinski, and Wojciech Ziarko. Rough sets. *Communications of the ACM*, 38(11):88–95, 1995.

[410] Zdzislaw Pawlak, Lech Polkowski, and Andrzej Skowron. Rough set theory. *KI*, 15(3):38–39, 2001.

[411] Zdzislaw Pawlak, S. K. Michael Wong, Wojciech Ziarko, et al. Rough sets: probabilistic versus deterministic approach. *International Journal of Man-Machine Studies*, 29(1):81–95, 1988.

[412] Michael Peer, Iva K. Brunec, Nora S. Newcombe, and Russell A. Epstein. Structuring knowledge with cognitive maps and cognitive graphs. *Trends in Cognitive Sciences*, 25:37–54, 2020.

[413] Xindong Peng and Hai-Hui Huang. Fuzzy decision making method based on cocoso with critic for financial risk evaluation. *Technological and Economic Development of Economy*, 2020.

[414] Dewayne E Perry and Alexander L Wolf. Foundations for the study of software architecture. *ACM SIGSOFT Software engineering notes*, 17(4):40–52, 1992.

[415] G. Pesant and Michel Gendreau. A constraint programming framework for local search methods. *Journal of Heuristics*, 5:255–279, 1999.

[416] Mario Petrich. On ideals of a semilattice. *Czechoslovak Mathematical Journal*, 22(3):361–367, 1972.

[417] Benjamin C Pierce. *Types and programming languages*. MIT press, 2002.

[418] Richard S Pierce. A note on complete boolean algebras. *Proceedings of the American Mathematical Society*, 9(6):892–896, 1958.

[419] Charles C. Pinter. A book of abstract algebra. 1982.

[420] Peter L Pirolli and John R Anderson. The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 39(2):240, 1985.

[421] Surapati Pramanik, Sourendranath Chackrabarti, Nandalal Ghosh, and Kumar Ashutosh. A study on problems of construction workers in west bengal based on neutrosophic cognitive maps. *International Journal of Innovative Research in Science, Engineering and Technology*, 2:6387–6394, 2014.

[422] Daniele Pretolani. Finding hypernetworks in directed hypergraphs. *European Journal of Operational Research*, 230(2):226–230, 2013.

[423] John G. Proakis. Probability, random variables and stochastic processes. *IEEE Trans. Acoust. Speech Signal Process.*, 33:1637, 1985.

[424] Kyoto Protocol. United nations framework convention on climate change. *Kyoto Protocol, Kyoto*, 19(8):1–21, 1997.

[425] Vladimir Semenovich Pugachev. Theory of random functions: And its application to control problems. 2013.

[426] Derek S Pugh, David J Hickson, Christopher R Hinings, and Christopher Turner. Dimensions of organization structure. *Administrative science quarterly*, pages 65–105, 1968.

[427] R Punitha and S Fathima Bazeela. Soft hyperfilters in hyperlattices. 2019.

[428] Madan Lal Puri and Dan A. Ralescu. The concept of normality for fuzzy random variables. *Annals of Probability*, 13:1373–1379, 1985.

[429] Madan Lal Puri and Dan A. Ralescu. Fuzzy random variables. *Journal of Mathematical Analysis and Applications*, 114:409–422, 1986.

[430] Mohan S Putcha. Semilattice decompositions of semigroups. In *Semigroup Forum*, volume 6, pages 12–34. Springer, 1973.

[431] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. *ArXiv*, abs/2003.07082, 2020.

[432] Zhi qiang Liu and Richard M. Satur. Contextual fuzzy cognitive map for decision support in geographic information systems. *IEEE Trans. Fuzzy Syst.*, 7:495–507, 1999.

[433] Dunwang Qin, Zhen Peng, and Lifeng Wu. Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction. *Knowl. Based Syst.*, 275:110700, 2023.

[434] Atiqe Ur Rahman, Muhammad Haris Saeed, and Hamiden Abd El-Wahed Khalifa. Multi-attribute decision-making based on aggregations and similarity measures of neutrosophic hypersoft sets with possibility setting. *Journal of Experimental & Theoretical Artificial Intelligence*, 36:161 – 186, 2022.

[435] George Rahonis. Fuzzy languages. In *Handbook of Weighted Automata*, pages 481–517. Springer, 2009.

[436] D. Rajan. Probability, random variables, and stochastic processes. 2017.

[437] Nadesan Ramaruban. *Commutative hyperalgebra*. University of Cincinnati, 2014.

[438] James O. Ramsay. Estimating smooth monotone functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 1998.

[439] Kristian Rautiainen, Joachim von Schantz, and Jarno Vähäniitty. Supporting scaling agile with portfolio management: Case paf.com. *2011 44th Hawaii International Conference on System Sciences*, pages 1–10, 2011.

[440] Victor J. Rayward-Smith. Hypergrammars: an extension of macrogrammars. *Journal of Computer and System Sciences*, 14(1):130–149, 1977.

[441] Lisbeth J Reales-Chacón, María E Lucena de Ustáriz, Francisco J Ustáriz-Fajardo, Andrea C Peñafiel Luna, Gabriela J Bonilla-Ayala, Pablo Djabayan-Djibeyan, José L Erazo-Parra, and Mónica A Valdiviezo-Maygua. Study of the efficacy of neural mobilizations to improve sensory and functional responses of lower extremities in older adults with diabetic peripheral neuropathy using plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 74:1–12, 2024.

[442] Alfréd Rényi. Representations for real numbers and their ergodic properties. *Acta Mathematica Academiae Scientiarum Hungarica*, 8:477–493, 1957.

[443] Akbar Rezaei, Florentin Smarandache, and S. Mirvakili. Applications of (neutro/anti)sophications to semihypergroups. *Journal of Mathematics*, 2021.

[444] Peter Ring. Relativistic mean field theory in finite nuclei. *Progress in Particle and Nuclear Physics*, 37:193–263, 1996.

[445] R Hugo Alberto Rivera and Hugo Alberto. The concept of corporate strategy. *Innovar-revista De Ciencias Administrativas Y Sociales*, 19:167–169, 2009.

[446] Judith Roitman. *Introduction to modern set theory*, volume 8. John Wiley & Sons, 1990.

[447] Kenneth H. Rose. A guide to the project management body of knowledge-fifth edition. *Project Management Journal*, 44, 2013.

[448] Nikolas Rose. Government and control. *British journal of criminology*, 40(2):321–339, 2000.

[449] Ivo G Rosenberg. An algebraic approach to hyperalgebras. In *Proceedings of 26th IEEE International Symposium on Multiple-Valued Logic (ISMVL'96)*, pages 203–207. IEEE, 1996.

[450] Daniel Rubio. Beginning django. In *Apress*, 2017.

[451] Maria Santilli Ruggero and Thomas Vougiouklis. Hyperstructures in lie-santilli admissibility and isotheories. *Ratio Mathematica*, 33:151, 2017.

[452] Magdalena Rusch, Josef-Peter Schöggl, and Rupert J. Baumgartner. Application of digital technologies for sustainable product management in a circular economy - a review. *Business Strategy and the Environment*, 2021.

[453] Gerald E Sacks. *Higher recursion theory*, volume 2. Cambridge University Press, 2017.

[454] Jacques Sakarovitch. Elements of automata theory. 2009.

[455] Chitrasen Samantra. Decision-making in fuzzy environment. 2012.

[456] Alexei Samsonovich and Bruce L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, 17:5900 – 5920, 1997.

[457] Gennadi A. Sardanashvily. Lectures on differential geometry of modules and rings. *arXiv: Mathematical Physics*, 2009.

[458] P Sathya, Nivetha Martin, and Florentine Smarandache. Plithogenic forest hypersoft sets in plithogenic contradiction based multi-criteria decision making. *Neutrosophic Sets and Systems*, 73:668–693, 2024.

[459] Richard M. Satur and Zhi qiang Liu. A contextual fuzzy cognitive map framework for geographic information systems. *IEEE Trans. Fuzzy Syst.*, 7:481–494, 1999.

[460] H. R. Schober. Introduction to lattice dynamics. 1995.

[461] Ken Schwaber. Agile project management with scrum. 1980.

[462] Philip Schwed. Topics in field theory. 1948.

[463] David B. Scott, Ruel Vance Churchill, and James Ward Brown. Complex variables and applications. *The Mathematical Gazette*, 69:243, 1985.

[464] Ganeshsree Selvachandran and Abdul Razak Salleh. Hypergroup theory applied to fuzzy soft sets. *Global Journal of Pure and Applied Sciences*, 11:825–834, 2015.

[465] Amartya Sen. Collective choice and social welfare. 2017.

[466] S. Shabalala. Product/business portfolio management: Logic, structural, and managerial implications. *South African Journal of Business Management*, 16:18–26, 1985.

[467] Sundas Shahzadi, Areen Rasool, Musavarah Sarwar, and Muhammad Akram. A framework of decision making based on bipolar fuzzy competition hypergraphs. *Journal of Intelligent & Fuzzy Systems*, 41(1):1319–1339, 2021.

[468] Francina Shalini. Trigonometric similarity measures of pythagorean neutrosophic hypersoft sets. *Neutrosophic Systems with Applications*, 2023.

[469] Iman Shames and Tyler Holt Summers. Network design via modular set function optimization in localization problems. *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 29–32, 2014.

[470] Iman Shames and Tyler Holt Summers. Rigid network design via submodular set function optimization. *IEEE Transactions on Network Science and Engineering*, 2:84–96, 2015.

[471] Iman Shames and Tyler Holt Summers. Corrections to "rigid network design via submodular set function optimization". *IEEE Transactions on Network Science and Engineering*, 7:2114–2114, 2020.

[472] John C Shepherdson and Howard E Sturgis. Computability of recursive functions. *Journal of the ACM (JACM)*, 10(2):217–255, 1963.

[473] Min Shi, Qunjing Wang, Guoli Li, Jiazi Xu, Qiyue Han, and Qiubo Ye. A new adaptive analytical model for the spherical reluctance motor based on hybrid trigonometric function-power function. *IEEE Transactions on Industrial Electronics*, 70:6099–6109, 2023.

[474] Patricia M. Shields, Nandhini Rangarajan, and Ok Stillwater. A playbook for research methods: Integrating conceptual frameworks and project management. 2013.

[475] David Shirley. Program management. *Project Management for Healthcare*, 2020.

[476] Thomas S. Shively, Thomas Sager, and Stephen G. Walker. A bayesian approach to non-parametric monotone function estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71, 2009.

[477] Roman Sikorski et al. *Boolean algebras*, volume 2. Springer, 1969.

[478] Seema Singh, DS Hooda, and SC Malik. Fuzzy rough information measures and their applications. *Int J Comput Sci Inf Technol (IJCSITY)*, 9(4):1–17, 2021.

[479] Sven Skyum and Leslie G. Valiant. A complexity theory based on boolean algebra. *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, pages 244–253, 1981.

[480] F. Smarandache. Introduction to superhyperalgebra and neutrosophic superhyperalgebra. *Journal of Algebraic Hyperstructures and Logical Algebras*, 2022.

[481] Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In *Philosophy*, pages 1–141. American Research Press, 1999.

[482] Florentin Smarandache. Neutrosophic set-a generalization of the intuitionistic fuzzy set. *International journal of pure and applied mathematics*, 24(3):287, 2005.

[483] Florentin Smarandache. Neutrosophic set–a generalization of the intuitionistic fuzzy set. *Journal of Defense Resources Management (JoDRM)*, 1(1):107–116, 2010.

[484] Florentin Smarandache. *Introduction to neutrosophic measure, neutrosophic integral, and neutrosophic probability*. Infinite Study, 2013.

[485] Florentin Smarandache. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*. Infinite study, 2018.

[486] Florentin Smarandache. Plithogeny, plithogenic set, logic, probability, and statistics. *arXiv preprint arXiv:1808.03948*, 2018.

[487] Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.

[488] Florentin Smarandache. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*. Infinite Study, 2020.

[489] Florentin Smarandache. History of superhyperalgebra and neutrosophic superhyperalgebra (revisited again). *Neutrosophic Algebraic Structures and Their Applications*, page 10, 2022.

[490] Florentin Smarandache. *Introduction to the n-SuperHyperGraph-the most general form of graph today*. Infinite Study, 2022.

[491] Florentin Smarandache. *Practical applications of IndetermSoft Set and IndetermHyperSoft Set and introduction to TreeSoft Set as an extension of the MultiSoft Set*. Infinite Study, 2022.

[492] Florentin Smarandache. *The SuperHyperFunction and the Neutrosophic SuperHyperFunction (revisited again)*, volume 3. Infinite Study, 2022.

[493] Florentin Smarandache. Decision making based on valued fuzzy superhypergraphs. 2023.

[494] Florentin Smarandache. Foundation of the superhypersoft set and the fuzzy extension superhypersoft set: A new vision. *Neutrosophic Systems with Applications*, 11:48–51, 2023.

[495] Florentin Smarandache. New types of topologies and neutrosophic topologies. *Neutrosophic Systems with Applications*, 1:1–3, 2023.

[496] Florentin Smarandache. New types of topologies and neutrosophic topologies (improved version). *Neutrosophic Sets and Systems*, 57(1):14, 2023.

[497] Florentin Smarandache. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*. Infinite Study, 2023.

[498] Florentin Smarandache. The cardinal of the m-powerset of a set of n elements used in the superhyperstructures and neutrosophic superhyperstructures. *Systems Assessment and Engineering Management*, 2:19–22, 2024.

[499] Florentin Smarandache. Foundation of revolutionary topologies: An overview, examples, trend analysis, research issues, challenges, and future directions. 2024.

[500] Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

[501] Florentin Smarandache. Superhyperstructure & neutrosophic superhyperstructure, 2024. Accessed: 2024-12-01.

[502] Florentin Smarandache and NM Gallup. Generalization of the intuitionistic fuzzy set to the neutrosophic set. In *International Conference on Granular Computing*, pages 8–42. Citeseer, 2006.

[503] Florentin Smarandache and Nivetha Martin. *Plithogenic n-super hypergraph in novel multi-attribute decision making*. Infinite Study, 2020.

[504] Florentin Smarandache, Memet Şahin, Derya Bakbak, Vakkas Uluçay, and Abdullah Kargın. *Neutrosophic SuperHyperAlgebra and New Types of Topologies*. Infinite Study, 2023.

[505] Florentin Smarandache, A. Saranya, A. Kalavathi, and S. Krishnaprakash. Neutrosophic superhypersoft sets. *Neutrosophic Sets and Systems*, 77:41–53, 2024.

[506] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 deep learning and unsupervised feature learning workshop*, volume 2010, pages 1–9. Vancouver, 2010.

[507] A Soltani Lashkenari and B Davvaz. Complete join hyperlattices. *Indian Journal of Pure and Applied Mathematics*, 46:633–645, 2015.

[508] Seok-Zun Song, Seon Jeong Kim, and Young Bae Jun. Hyperfuzzy ideals in bck/bci-algebras. *Mathematics*, 5(4):81, 2017.

[509] Harold C Sox, Michael C Higgins, Douglas K Owens, and Gillian Sanders Schmidler. *Medical decision making*. John Wiley & Sons, 2024.

[510] Ieee Std. Ieee guide adoption of pmi standard a guide to the project management body of knowledge. 2004.

[511] Elias M. Stein and Timothy S. Murphy. Harmonic analysis: Real-variable methods, orthogonality, and oscillatory integrals. 1993.

[512] Mikhail A. Stephanov, Jacobus J. M. Verbaarschot, and Tilo Wettig. Random matrices. 2005.

[513] W. Stephenson. Modules whose lattice of submodules is distributive. *Proceedings of The London Mathematical Society*, pages 291–310, 1974.

[514] Christoph Johann Stettina and Jeannette Hörz. Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management*, 33:140–152, 2015.

[515] Inssaf Azouagh PhD Student and Driss Daoui PhD Professor. Comparative analysis of active and passive portfolio management: A theoretical approach analyse comparative de la gestion active et passive des portefeuilles : Une approche théorique. 2023.

[516] John Brendan Sullivan. Representations of the hyperalgebra of an algebraic group. *American Journal of Mathematics*, 100(3):643–652, 1978.

[517] Libor Svadlenka, Vladimir Simic, Momcilo Dobrodolac, Dragana Lazarevic, and Gordana Todorovic. Picture fuzzy decision-making approach for sustainable last-mile delivery. *IEEE Access*, 8:209393–209414, 2020.

[518] Madeline Al Tahan, Sarka Hoskova-Mayerova, and Bijan Davvaz. An overview of topological hypergroupoids. *J. Intell. Fuzzy Syst.*, 34:1907–1916, 2018.

[519] Siu-Lan Tan, Elizabeth M Wakefield, and Paul W. Jeffries. Musically untrained college students' interpretations of musical notation: sound, silence, loudness, duration, and temporal order. *Psychology of Music*, 37:24 – 5, 2009.

[520] Bewket Tadesse Tiruneh, Gayle McLelland, and Virginia Plummer. National healthcare system development of ethiopia: A systematic narrative review. *Hospital Topics*, 98:37 – 44, 2020.

[521] Donald M. Topkis. Minimizing a submodular function on a lattice. *Oper. Res.*, 26:305–321, 1978.

[522] Alberto Torchinsky. Real-variable methods in harmonic analysis. 1986.

[523] Joel A. Tropp. An introduction to matrix concentration inequalities. *ArXiv*, abs/1501.01571, 2015.

[524] Antonia Maria Tulino and Sergio Verdú. Random matrix theory and wireless communications. *Found. Trends Commun. Inf. Theory*, 1, 2004.

[525] Michael L Tushman and David A Nadler. Information processing as an integrating concept in organizational design. *Academy of management review*, 3(3):613–624, 1978.

[526] Vasileios Tzoumas. Resilient submodular maximization for control and sensing. 2018.

[527] Vasileios Tzoumas, Ali Jadbabaie, and George Pappas. Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms. *2016 American Control Conference (ACC)*, pages 191–196, 2015.

[528] Muhammad Umair, Syed Jalaluddin Hashmi, and Young-Koo Lee. Optimizing keyphrase ranking for relevance and diversity using submodular function optimization (sfo). *ArXiv*, abs/2410.20080, 2024.

[529] Russell R. Vane. Advances in hypergame theory. 2006.

[530] M. Venkatesh and Dr. K. Thangadurai. Survey for fuzzy-contextual contrast enhancement. 2017.

[531] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, 2010.

[532] Oleg Ya. Viro. Hyperfields for tropical geometry i. hyperfields and dequantization. *arXiv: Algebraic Geometry*, 2010.

[533] Ashwin Vishwanathan, Alexandro D. Ramirez, Jingpeng Wu, Alexander Sood, Runzhe Yang, Nico Kemnitz, Dodam Ih, Nicholas L. Turner, Kisuk Lee, Ignacio Tartavull, William M. Silversmith, Chris S. Jordan, Celia David, Doug Bland, Mark S. Goldman, Emre R. F. Aksay, and H. Sebastian Seung. Predicting modular functions and neural coding of behavior from a synaptic wiring diagram. *bioRxiv*, 2020.

[534] Souzana Vougioukli. Helix hyperoperation in teaching research. *Science & Philosophy*, 8(2):157–163, 2020.

[535] Souzana Vougioukli. Hyperoperations defined on sets of s -helix matrices. 2020.

[536] Souzana Vougioukli. Helix-hyperoperations on lie-santilli admissibility. *Algebras Groups and Geometries*, 2023.

[537] T Vougiouklis. The set of hypergroups with operators which are constructed from a set with two elements. *Acta Universitatis Carolinae. Mathematica et Physica*, 22(1):7–10, 1981.

[538] T Vougiouklis. The santilli's theory 'invasion'in hyperstructures. *Algebras, Groups and Geometries*, 28(1):83–103, 2011.

[539] T Vougiouklis. Fundamental relations in hv-structures. the'judging from the results' proof. *Journal of Algebraic Hyperstructures and Logical Algebras*, 1(1):21–36, 2020.

[540] Thomas Vougiouklis. On some representations of hypergroups. *Annales scientifiques de l'Université de Clermont. Mathématiques*, 95(26):21–29, 1990.

[541] THOMAS Vougiouklis. The fundamental relation in hyperrings. the general hyperfield. In *Proc. Fourth Int. Congress on Algebraic Hyperstructures and Applications (AHA 1990), World Scientific*, pages 203–211. World Scientific, 1991.

[542] Thomas Vougiouklis. Hv-groups defined on the same set. *Discrete Mathematics*, 155(1-3):259–265, 1996.

[543] Thomas Vougiouklis. Hypermathematics, hv-structures, hypernumbers, hypermatrices and lie-santilli admissibility. *American Journal of Modern Physics*, 4(5):38–46, 2015.

[544] Victor Harold Vroom and Arthur G. Jago. Leadership and decision making. *Leadership*, 2018.

[545] Iain Wallace. The global economic system. 1989.

[546] Rosanna Walters-Symons, Mark Wilson, Andre Klostermann, and Samuel Vine. Examining the response programming function of the quiet eye: Do tougher shots need a quieter eye? *Cognitive processing*, 19:47–52, 2018.

[547] Juntao Wang, Youngbae Jun, Xiaolong Xin, and Yuxi Zou. On derivations of bounded hyperlattices. *Journal of mathematical research with applications*, 36(2):151–161, 2016.

[548] Meng Wang and Fanghui Hu. The application of nltk library for python natural language processing in corpus research. *Theory and Practice in Language Studies*, 2021.

[549] Yue Wang, Hung Le, Akhilesh Deepak Gotmare, Nghi D. Q. Bui, Junnan Li, and Steven C. H. Hoi. Codet5+: Open code large language models for code understanding and generation. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[550] Chris Wendler, Markus Püschel, and Dan Alistarh. Powerset convolutional neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[551] Andrew M. Wikenheiser and Geoffrey Schoenbaum. Over the river, through the woods: cognitive maps in the hippocampus and orbitofrontal cortex. *Nature Reviews Neuroscience*, 17:513–523, 2016.

[552] Rebecca Willis. The role of national politicians in global climate governance. *Environment and Planning E: Nature and Space*, 3:885 – 903, 2020.

[553] Zoie Shui-Yee Wong and Michael L. Rigby. Identifying and addressing digital health risks associated with emergency pandemic response: Problem identification, scoping review, and directions toward evidence-based evaluation. *International Journal of Medical Informatics*, 157:104639 – 104639, 2021.

[554] Felix Wu, Tianyi Zhang, Amauri H. de Souza, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, 2019.

[555] Jie Wu. Small worlds: the dynamics of networks between order and randomness. *SIGMOD Rec.*, 31:74–75, 2000.

[556] Iannis Xenakis and Sharon Kanach. Formalized music: Thought and mathematics in composition. 1971.

[557] Liming Xiao, Guangquan Huang, Witold Pedrycz, Dragan Pamucar, Luis Martínez, and Genbao Zhang. A q-rung orthopair fuzzy decision-making model with new score function and best-worst method for manufacturer selection. *Information sciences*, 608:153–177, 2022.

[558] Xiaolong Xin, Pengfei He, and Yongwei Yang. Fuzzy soft hyperideals in hyperlattices. *Politehn. Univ. Bucharest Sci. Bull. Ser. A Appl. Math. Phys*, 77:173–184, 2015.

[559] Xinzheng Xu, Xiaoyang Zhao, Meng Wei, and Zhongnian Li. A comprehensive review of graph convolutional networks: approaches and applications. *Electronic Research Archive*, 2023.

[560] Yang Xu, Da Ruan, KY Qin, and Jun Liu. Lattice-valued logic. *Studies in fuzziness and soft computing*, 132, 2003.

[561] Zenglin Xu, Bin Liu, Shandian Zhe, Haoli Bai, Zihan Wang, and Jennifer Neville. Variational random function model for network modeling. *IEEE Transactions on Neural Networks and Learning Systems*, 30:318–324, 2019.

[562] Jian Yang and Joseph Y-T Leung. A generalization of the weighted set covering problem. *Naval Research Logistics (NRL)*, 52(2):142–149, 2005.

[563] Kai Hao Yang and Alexander Zentefis. Monotone function intervals: Theory and applications. *American Economic Review*, 2023.

[564] Rui Yang, Qingcheng Zeng, Keen You, Yujie Qiao, Lucas Huang, Chia-Chun Hsieh, Benjamin Rosand, Jeremy Goldwasser, Amisha D. Dave, Tiarnán D. L. Keenan, Emily Y. Chew, Dragomir R. Radev, Zhiyong Lu, Hua Xu, Qingyu Chen, and Irene Li. Ascle: A python natural language processing toolkit for medical text generation. 2023.

[565] Edwin Engin Yaz. Linear matrix inequalities in system and control theory. *Proceedings of the IEEE*, 86:2473–2474, 1998.

[566] Morteza Yazdani, Ali Ebadi Torkayesh, ?eljko Stević, Prasenjit Chatterjee, Sahand Asgharieh Ahari, and Violeta Doval Hernandez. An interval valued neutrosophic decision-making structure for sustainable supplier selection. *Expert Syst. Appl.*, 183:115354, 2021.

[567] Yunqiang Yin, Jianming Zhan, Dehua Xu, and Jiayin Wang. The l-fuzzy hypermodules. *Comput. Math. Appl.*, 59:953–963, 2010.

[568] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[569] Lotfi A Zadeh. A fuzzy-set-theoretic interpretation of linguistic hedges. 1972.

[570] Lotfi A Zadeh. Fuzzy sets and their application to pattern classification and clustering analysis. In *Classification and clustering*, pages 251–299. Elsevier, 1977.

[571] Lotfi A Zadeh. Fuzzy sets versus probability. *Proceedings of the IEEE*, 68(3):421–421, 1980.

[572] Lotfi A Zadeh. Fuzzy logic, neural networks, and soft computing. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 775–782. World Scientific, 1996.

[573] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[574] Team Glm Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Ming yue Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiaoyu Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yi An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhenyi Yang, Zhengxiao Du, Zhen-Ping Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *ArXiv*, abs/2406.12793, 2024.

[575] Jianming Zhan, Bijan Davvaz, and Kar Ping Shum. A new view on fuzzy hypermodules. *Acta Mathematica Sinica, English Series*, 23:1345–1356, 2007.

[576] Mark Zhandry. How to construct quantum random functions. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, 2012.

[577] Jing Zhang. A study on mental health assessments of college students based on triangular fuzzy function and entropy weight method. *Mathematical Problems in Engineering*, 2021.

[578] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6, 2019.

[579] Tao Zhang, Gang Li, T. C. Edwin Cheng, and Stephen Shum. Consumer inter-product showrooming and information service provision in an omni-channel supply chain. *Decis. Sci.*, 51:1232–1264, 2019.

[580] Weijun Zhou, Donghui Li, Weijun Zhou, and LI D.H. Limited memory bfgs method for nonlinear monotone equations. 2007.

[581] Norman A Zimmerman. System integration as a programming function. In *Proceedings of the 1969 24th national conference*, pages 459–467, 1969.

[582] Yu. F. Zin'kovskii and B. M. Uvarov. Hyper-randomness of algorithms for simulation of current radio electronic equipment. *Radioelectronics and Communications Systems*, 54:147–154, 2011.

[583] William S. Zwicker. Playing games with games: the hypergame paradox. *American Mathematical Monthly*, 94:507–514, 1987.