Introduction for Structure, HyperStructure, SuperHyperStructure, MultiStructure, Iterative MultiStructure, TreeStructure, and ForestStructure

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

Abstract

We introduce a unifying framework for algebraic and relational models by defining *Structure* as a nonempty carrier set equipped with one or more operations satisfying prescribed axioms. We then develop its hierarchical extensions: *HyperStructure* and *SuperHyperStructure*, which lift operations to iterated powersets and hyperoperations; *MultiStructure* and *Iterative MultiStructure*, which employ multisets and their recursive application; and *TreeStructure* and *ForestStructure*, which organize operations over rooted trees and forests of attributes. To encompass even broader constructions we introduce *Any-Structure*, capturing arbitrary compositions of such functorial constructors, and *U-Structure*, which enriches any carrier with degrees of uncertainty drawn from a chosen uncertainty model. We present *Functorial Structure* as the category-theoretic abstraction underpinning all these notions. Furthermore, we examine the notions of Curried Structure and Dynamic Structure. Together, these concepts generalize classical frameworks in topology, graph theory, automata, lattice and group theory, and provide a coherent foundation for modeling complex, hierarchical, and uncertain systems.

Keywords: HyperStructure, SuperHyperStructure, Treestructure, Foreststructure, MultiStructure, Iterative MultiStructure

Structure of this paper

The format of this paper is described below.

1 Classical Structure and HyperStructure

A *Classical Structure* denotes any mathematical or real-world concept—examples include logic, probability, statistics, algebra, geometry, graphs, and automata. A *HyperStructure* extends this notion by replacing a base set S with its powerset $\mathcal{P}(S)$ and equipping it with hyperoperations that combine subsets into subsets, thereby enabling higher-order relations [1–6]. All structures considered in this paper are assumed to be finite. Definitions are given below.

Definition 1.1 (Classical Structure)**.** A *Classical Structure* C is a mathematical structure drawn from one of various domains—such as Set theory, Logic, Probability, Statistics, Algebra, Geometry, Graph theory, Automata theory, Game theory, etc.—and can be formalized as a pair

$$C = \big(H, \{\#^{(m)}\}_{m \in \mathcal{I}}\big),$$

where:

- H is a nonempty set (the *carrier* or *universe*).

- For each $m \in \mathcal{I} \subseteq \mathbb{Z}_{>0}$, there is an m-ary operation

$$\#^{(m)} \ : \ H^m \ \longrightarrow \ H,$$

subject to specified *axioms* (e.g., associativity, commutativity, identity laws) depending on the particular type of structure.

We say that C is of *type* $\{\#^{(m)} : m \in \mathcal{I}\}$. Examples include:

- A *Set* $(S, \ \emptyset)$ viewed as a carrier with distinguished elements or relations, but no additional operations [7].
- A *Logic* $(L, \ \wedge, \vee, \neg)$, where $\wedge$ and $\vee$ are binary connectives and $\neg$ is a unary connective satisfying logical axioms [8].
- A *Probability* structure $(\Omega, \mathcal{F}, P)$, where $P : \mathcal{F} \to [0, 1]$ is a measure on a sigma-algebra $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ [9–11].
- A *Statistics* model $(X, \mathcal{A}, \theta)$, where θ maps data X to statistical parameters [12].
- A *Algebraic* structure:
  - A *Group* $(G, *)$ with $* : G \times G \to G$ satisfying associativity, identity, and inverses [13, 14].
  - A *Ring* $(R, +, \times)$ with two binary operations satisfying ring axioms [15, 16].
  - A *Vector Space* $(V, +, \cdot)$ over a field $\mathbb{F}$, where $\cdot : \mathbb{F} \times V \to V$ [17, 18].
- A *Geometric* structure $(X, \ \mathrm{dist})$, where $\mathrm{dist} : X \times X \to \mathbb{R}$ satisfies the metric axioms [19, 20].
- A *Graph* (V, E), where $E \subseteq \{\{u, v\} \mid u, v \in V\}$ in the undirected case (or $E \subseteq V \times V$ in the directed case), with additional notions of adjacency and incidence [21, 22].
- An *Automaton* $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the set of accepting states [23, 24].
- A *Game* $(N, \{A_i\}, \{u_i\})$, where N is a set of players, A_i is the action set for player i, and $u_i : \prod_{j \in N} A_j \to \mathbb{R}$ is the utility (payoff) function for player i [25, 26].

Definition 1.2 (Powerset)**.** (cf. [27, 28]) For any set S, its *powerset* $\mathcal{P}(S)$ is the collection of all subsets of S, including $\emptyset$ and S itself:

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

Definition 1.3 (Hyperoperation)**.** *(cf. [29–32])* A *hyperoperation* on a set S is a binary rule whose output is a subset of S rather than a single element. Formally, it is a map

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

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

Definition 1.4 (Hyperstructure)**.** *(cf. [33–38])* A *Hyperstructure* generalizes a Classical Structure by replacing the base set S with its powerset. It is given by

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

where $\circ$ is a hyperoperation acting on subsets of S. This framework allows operations to combine collections of elements into other collections.

Example 1.5 (Beverage-Mixing Hyperstructure)**.** Let

$$S = \{\mathsf{Sugar}, \ \mathsf{Lemon}, \ \mathsf{Water}, \ \mathsf{Ice}, \ \mathsf{Mint}\}$$

be a set of basic drink ingredients. Define a binary *hyperoperation*

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

by specifying for each ordered pair $(a, b) \in S \times S$ the set of possible combined ingredients:

$$\begin{aligned} &\textsf{Sugar} \circ \textsf{Lemon} = \{\textsf{Lemonade}, \ \textsf{SweetLemonade}\}, \\ &\textsf{Water} \circ \textsf{Mint} = \{\textsf{MintWater}\}, \\ &\textsf{Lemon} \circ \textsf{Ice} = \{\textsf{IcedLemonade}\}, \quad \text{etc.} \end{aligned}$$

We extend this to all subsets $A, B \subseteq S$ by

$$A \circ B \; = \bigcup_{a \in A,\; b \in B} (a \circ b) \;\subseteq\; S.$$

Then $\mathcal{H} = (\mathcal{P}(S), \circ)$ is a Hyperstructure: each pair of ingredient-sets combines into a new set of possible drink recipes. This models the real-world process of mixing any collection of ingredients into an array of beverage variants, capturing the non-deterministic nature of recipe creation.

2 SuperHyperStructure

A *SuperHyperStructure* extends a HyperStructure by applying the powerset construction recursively n times. Its operations act on nested collections to model hierarchical, multi-level interactions and relations [39–44]. Related concepts include SuperHyperAlgebra [45–48], SuperHyperGraph [49–55], and other super-level algebraic and graph-theoretic structures.

Definition 2.1 (n-th Powerset)**.** [39, 40, 56–58] Let H be a set. Define inductively

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

Then $\mathcal{P}^1(H) = \mathcal{P}(H)$, $\mathcal{P}^2(H) = \mathcal{P}(\mathcal{P}(H))$, and so on. Removing the empty set at each stage yields the *nonempty* iterated powersets. Writing $\mathcal{P}^*(X) = \mathcal{P}(X) \setminus \{\emptyset\}$, we set

$$\mathcal{P}^{*1}(H) = \mathcal{P}^*(H), \quad \mathcal{P}^{*(k+1)}(H) = \mathcal{P}^*\big(\mathcal{P}^{*k}(H)\big).$$

Example 2.2 (Travel Planning via the n-th Powerset)**.** Let

$$H = \{\textsf{Flight}, \ \textsf{Hotel}, \ \textsf{CarRental}\}$$

be the basic travel services. Then:

- The first powerset $\mathcal{P}^1(H) = \mathcal{P}(H)$ consists of all possible service-sets:

  $$\{\}, \ \{\textsf{Flight}\}, \ \{\textsf{Hotel}\}, \ \{\textsf{CarRental}\}, \ \{\textsf{Flight}, \textsf{Hotel}\}, \ \dots, \ \{\textsf{Flight}, \textsf{Hotel}, \textsf{CarRental}\}.$$

  For example, $\{\textsf{Flight}, \textsf{Hotel}\}$ is a "flight + hotel" package.
- The second powerset $\mathcal{P}^2(H) = \mathcal{P}(\mathcal{P}(H))$ is the set of all collections of such packages. Concretely:

  $$\big\{\{\textsf{Flight}, \textsf{Hotel}\}, \ \{\textsf{CarRental}\}\big\} \quad \text{(two-option bundle)},$$

  $$\big\{\{\textsf{Flight}\}\big\} \quad \text{(single-package offer)},$$

  $$\big\{\{\textsf{Hotel}\}, \ \{\textsf{Flight}, \textsf{CarRental}\}\big\}, \quad \text{etc.}$$
- The third powerset $\mathcal{P}^3(H) = \mathcal{P}(\mathcal{P}^2(H))$ consists of all collections of these bundles. For instance:

  $$\Big\{\{\{\textsf{Flight}, \textsf{Hotel}\}, \{\textsf{CarRental}\}\}, \ \{\{\textsf{Flight}\}\}\Big\}$$

  might represent a meta-offer that groups two distinct bundle-offers for sale.

Thus $\mathcal{P}^n(H)$ encodes *n-level* groupings of travel packages: from individual services, to packages, to bundles of packages, and so on.

Definition 2.3 (SuperHyperOperations)**.** *(cf. [33])* Let H be a nonempty set and define its iterated powersets by

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

An (m, n)-*SuperHyperOperation* is then an m-ary map

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

where $\mathcal{P}^n_*(H)$ is either the full n-th powerset $\mathcal{P}^n(H)$ or its nonempty version $\mathcal{P}^n(H) \setminus \{\emptyset\}$. When the codomain excludes $\emptyset$ it is called a *classical-type* SuperHyperOperation, and when it includes $\emptyset$ it is termed a *Neutrosophic* SuperHyperOperation.

Definition 2.4 (n-Superhyperstructure)**.** *(cf. [33, 38, 40, 59])* An *n-Superhyperstructure* builds on a Hyperstructure by using the n-th iterated powerset of the base set S. It is presented as

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

where $\circ$ is an operation defined on $\mathcal{P}^n(S)$. This construction yields a hierarchy of operations acting on increasingly nested collections of elements.

Example 2.5 (Consultancy Task Allocation as a 2-Superhyperstructure)**.** Let

$$H = \{\text{Masafumi, Yutaka, Tae}\}$$

be a team of consultants. We take $n = 2$, so that

$$\mathcal{P}^2(H) = \mathcal{P}\big(\mathcal{P}(H)\big) = \{\, C \mid C \subseteq \mathcal{P}(H)\},$$

the collection of all sets of consultant-subsets. We define a $(2, 2)$-SuperHyperOperation

$$\circ^{(2,2)} : H \times H \longrightarrow \mathcal{P}^2_*(H)$$

by, for any distinct $x, y \in H$,

$$\circ^{(2,2)}(x, y) = \big\{\{\, \{x\}, \{y\}\}, \ \{\{x, y\}\}\big\} \in \mathcal{P}^2_*(H),$$

and $\circ^{(2,2)}(x, x) = \{\{\{x\}\}\}$. Here:

- $\{\{x\}, \{y\}\}$ represents assigning x and y to two separate individual projects.
- $\{\{x, y\}\}$ represents assigning x and y together to a joint project.

Thus the 2-Superhyperstructure

$$\mathcal{SH}_2 = \big(\mathcal{P}^2(H), \circ^{(2,2)}\big)$$

models the two-level task-allocation process in the consultancy: first, consultants combine into teams (level 1 subsets in $\mathcal{P}(H)$), and second, the operation $\circ^{(2,2)}$ yields a set of possible team-of-teams assignments (elements of $\mathcal{P}^2(H)$).

3 Weak HyperStructure and Weak SuperHyperStructure

A weak hyperstructure (Hv–structure) is an algebraic system whose hyperoperation satisfies weak associativity: any two ways of bracketing yield intersecting product sets [60–63].

Definition 3.1 (Weak Hyperstructure (Hv–structure))**.** [34] Let H be a nonempty set and let

$$\cdot : H \times H \longrightarrow \mathcal{P}^*(H)$$

be a *hyperoperation*, where $\mathcal{P}^*(H) = \mathcal{P}(H) \setminus \{\varnothing\}$ is the set of all nonempty subsets of H. We say:

1. $\cdot$ is *weakly associative* if

$$x \cdot \big(y \cdot z\big) \cap \big(x \cdot y\big) \cdot z \neq \varnothing \quad \forall\, x, y, z \in H.$$

2. $\cdot$ is *weakly commutative* if

$$x \cdot y \;\cap\; y \cdot x \;\neq\; \varnothing \quad \forall x, y \in H.$$

3. $(H, \cdot)$ is called an *Hv–semigroup* if $\cdot$ is weakly associative.

4. $(H, \cdot)$ is called an *Hv–group* if it is an Hv–semigroup and also satisfies the *reproduction axiom*:

$$a \cdot H \;=\; H \quad \text{and} \quad H \cdot a \;=\; H \quad \forall\, a \in H.$$

Example 3.2 (Chemical Reaction Hv–semigroup: Water Formation)**.** Consider the set of chemical species

$$H \;=\; \{\mathrm{H_2},\ \mathrm{O_2},\ \mathrm{H_2O}\}.$$

Define a hyperoperation $\cdot : H \times H \to \mathcal{P}^*(H)$ by specifying its values on pairs of basic species and extending to all nonempty subsets $A, B \subseteq H$ via

$$A \cdot B \;=\; \bigcup_{a \in A,\, b \in B} \bigl(a \cdot b\bigr).$$

On atoms, set:

$$\mathrm{H_2} \cdot \mathrm{O_2} \;=\; \{\mathrm{H_2O}\}, \quad \mathrm{O_2} \cdot \mathrm{H_2} \;=\; \{\mathrm{H_2O}\},$$
$$\mathrm{H_2O} \cdot \mathrm{H_2} \;=\; \{\mathrm{H_2O},\ \mathrm{H_2}\}, \quad \mathrm{H_2O} \cdot \mathrm{O_2} \;=\; \{\mathrm{H_2O},\ \mathrm{O_2}\},$$

and for each $x \in H,\ x \cdot x = \{x\}$.

One checks:

- *Weak associativity:* For every $x, y, z \in H$,

$$x \cdot \bigl(y \cdot z\bigr) \;\cap\; \bigl(x \cdot y\bigr) \cdot z \;=\; \{\mathrm{H_2O}\} \;\neq\; \varnothing.$$

  For example, $\mathrm{H_2} \cdot (\mathrm{O_2} \cdot \mathrm{H_2}) = \mathrm{H_2} \cdot \{\mathrm{H_2O}\} = \{\mathrm{H_2O}, \mathrm{H_2}\}$ and $(\mathrm{H_2} \cdot \mathrm{O_2}) \cdot \mathrm{H_2} = \{\mathrm{H_2O}\} \cdot \mathrm{H_2} = \{\mathrm{H_2O}, \mathrm{H_2}\}$.

- *Weak commutativity:* Since $a \cdot b = b \cdot a$ for all a, b, clearly

$$a \cdot b \;\cap\; b \cdot a \;=\; a \cdot b \;\neq\; \varnothing.$$

Therefore $(H, \cdot)$ is an *Hv–semigroup* modeling the (possibly reversible) reaction network among hydrogen, oxygen, and water.

A weak n-Superhyperstructure (SHv-Structure) equips the n-th powerset of a base set with a superhyperoperation satisfying weak associativity across hierarchical levels [64].

Definition 3.3 (Weak n^{th} Superhyperstructure)**.** [64] Let S be a nonempty set and let $\mathcal{P}^n(S)$ be its n^{th} powerset (Definition 2.1). Denote by $\mathcal{P}^*(\mathcal{P}^n(S)) = \mathcal{P}(\mathcal{P}^n(S)) \setminus \{\varnothing\}$ the collection of all nonempty subsets of $\mathcal{P}^n(S)$. A *weak n-Superhyperstructure* is a pair

$$\bigl(\mathcal{P}^n(S),\ \circ\bigr)$$

where

$$\circ :\ \mathcal{P}^n(S) \times \mathcal{P}^n(S) \;\longrightarrow\; \mathcal{P}^*(\mathcal{P}^n(S))$$

is a *superhyperoperation* satisfying the *weak associativity* condition:

$$X \circ \bigl(Y \circ Z\bigr) \;\cap\; \bigl(X \circ Y\bigr) \circ Z \;\neq\; \varnothing \quad \text{for all } X, Y, Z \in \mathcal{P}^n(S).$$

If moreover for each $A \in \mathcal{P}^n(S)$ one has $A \circ \mathcal{P}^n(S) = \mathcal{P}^n(S) = \mathcal{P}^n(S) \circ A$, then $(\mathcal{P}^n(S), \circ)$ is called an *Hv–group of order n*.

Example 3.4 (Weak 2^{nd} Superhyperstructure: Corporate Committee Formation)**.** Let the base set of employees be

$$S = \{\text{Masafumi}, \ \text{Yutaka}, \ \text{Tae}\}.$$

Then

$$\mathcal{P}^1(S) = \big\{\{\text{Masafumi}\}, \{\text{Yutaka}\}, \{\text{Tae}\}, \{\text{Masafumi, Yutaka}\}, \{\text{Masafumi, Tae}\}, \{\text{Yutaka, Tae}\}, S\big\},$$

and

$$\mathcal{P}^2(S) = \mathcal{P}\big(\mathcal{P}^1(S)\big)$$

is the set of all collections of such teams. Define three "committees" (elements of $\mathcal{P}^2(S)$):

$$X = \big\{\{\text{Masafumi, Yutaka}\}, \{\text{Yutaka, Tae}\}\big\},$$

$$Y = \big\{\{\text{Yutaka, Tae}\}, \{\text{Masafumi, Tae}\}\big\},$$

$$Z = \big\{\{\text{Masafumi, Yutaka}\}\big\}.$$

We introduce a superhyperoperation

$$\circ : \mathcal{P}^2(S) \times \mathcal{P}^2(S) \longrightarrow \mathcal{P}^*\big(\mathcal{P}^2(S)\big)$$

by

$$A \circ B = \big\{ A \cup B, \ \{\text{Masafumi, Yutaka, Tae}\}\big\}, \quad \forall A, B \in \mathcal{P}^2(S),$$

where $\{\text{Masafumi, Yutaka, Tae}\} \in \mathcal{P}^1(S) \subset \mathcal{P}^2(S)$ is the "full-staff committee."

Weak associativity:

$$Y \circ Z = \big\{ Y \cup Z, \ \{\text{Masafumi, Yutaka, Tae}\}\big\} = \big\{\{\text{Masafumi, Yutaka, Tae}\}\big\},$$

so

$$X \circ \big(Y \circ Z\big) = X \circ \{\{\text{Masafumi, Yutaka, Tae}\}\}$$
$$= \big\{ X \cup \{\text{Masafumi, Yutaka, Tae}\}, \{\text{Masafumi, Yutaka, Tae}\}\big\}$$
$$= \big\{\{\text{Masafumi, Yutaka, Tae}\}\big\}.$$

On the other hand,

$$X \circ Y = \big\{ X \cup Y, \ \{\text{Masafumi, Yutaka, Tae}\}\big\} \ni \{\text{Masafumi, Yutaka, Tae}\},$$

hence

$$\big(X \circ Y\big) \circ Z = \big\{\{\text{Masafumi, Yutaka, Tae}\}\big\} \circ Z$$
$$= \big\{\{\text{Masafumi, Yutaka, Tae}\}\big\}.$$

Therefore

$$X \circ \big(Y \circ Z\big) \ \cap \ \big(X \circ Y\big) \circ Z \ =$$
$$\{\{\text{Masafumi, Yutaka, Tae}\}\} \ \neq \ \emptyset,$$

verifying weak associativity. Thus $\big(\mathcal{P}^2(S), \circ\big)$ is a weak 2^{nd} Superhyperstructure modeling how sub-committees combine into higher-level oversight groups.

4 Tree Structure

A Tree Structure is a finite poset with a unique root and linearly ordered principal ideals, ensuring hierarchical parent–child relationships [65, 66].

Definition 4.1 (Tree Structure)**.** (cf. [65, 66]) Let T be a nonempty finite set, and let $\leq$ be a binary relation on T. We say that $(T, \leq)$ is a *Tree Structure* (more precisely, a finite rooted tree) if and only if the following conditions hold:

(i) $\leq$ is a partial order on T. In other words, for all $x, y, z \in T$:

- $x \leq x$ (reflexivity),
- If $x \leq y$ and $y \leq x$, then $x = y$ (antisymmetry),
- If $x \leq y$ and $y \leq z$, then $x \leq z$ (transitivity).

(ii) There exists a unique minimal element $r \in T$ (called the *root*) such that

$$\forall x \in T, \ r \leq x.$$

(iii) For every $x \in T$, the set

$$\{ y \in T \mid y \leq x \}$$

is totally ordered by $\leq$. That is, if $y_1, y_2 \in T$ and $y_1, y_2 \leq x$, then either $y_1 \leq y_2$ or $y_2 \leq y_1$.

In this situation, we call T a finite rooted tree under the order $\leq$, and refer to $(T, \leq)$ as a *Tree Structure*.

Definition 4.2 (Tree Operation)**.** (cf. [65, 66]) Let $(T, \leq)$ be a Tree Structure (as above). A *Tree Operation* of arity m (for some integer $m \geq 1$) is any function

$$\star : T^m \longrightarrow T.$$

Equivalently, a Tree Operation is an m-ary mapping that takes m nodes of the tree (possibly with repetition) and produces another node in T. We may impose additional properties on $\star$, such as:

- **Monotonicity:** $\star$ is called *monotone* (or order-preserving) if for all tuples

  $$(x_1, x_2, \ldots, x_m), \ (y_1, y_2, \ldots, y_m) \in T^m,$$

  whenever $x_i \leq y_i$ in T for each $i = 1, 2, \ldots, m$, it follows that

  $$\star(x_1, x_2, \ldots, x_m) \leq \star(y_1, y_2, \ldots, y_m) \quad \text{in } T.$$

- **Arity and Notation:** When $m = 1$, we often call $\star$ a *unary tree operation*; when $m = 2$, a *binary tree operation*, and so on. For a binary Tree Operation, it is customary to write

  $$x \star y = \star(x, y) \quad \big(x, y \in T\big).$$

In summary, a Tree Operation is simply a function from T^m to T, defined on a rooted-tree domain $(T, \leq)$. If no further axioms are imposed, any such mapping qualifies as a Tree Operation of arity m.

Example 4.3 (File System Directory Tree)**.** Let

$$T = \big\{\texttt{/},\ \texttt{/home},\ \texttt{/home/Masafumi},\ \texttt{/home/Masafumi/docs},\ \texttt{/var},\ \ldots\big\}$$

be the set of all directory paths on a single UNIX-style filesystem. Define the order $\leq$ by “path prefix”:

$$p \leq q \iff p \text{ is a prefix of } q \text{ (i.e. directory containment).}$$

Then “/” is the unique minimal element (the root), every prefix-chain $\{ p \mid p \leq q\}$ is totally ordered by containment, and $\leq$ is clearly a partial order. Thus $(T, \leq)$ is a Tree Structure modeling the directory hierarchy.

Example 4.4 (Corporate Management Tree)**.** Let

$$T = \{\text{CEO, VP}_1, \text{VP}_2, \text{Mgr}_A, \text{Emp}_A 1, \text{Emp}_A 2, \dots\}$$

be the set of employees in one corporation. Define

$$x \ \leq \ y \quad \iff \quad x \text{ is on the chain of command from } y \text{ up to the CEO.}$$

Then "CEO" is the unique minimal element, each employee's chain to the CEO is totally ordered, and the relation is reflexive, antisymmetric, and transitive. Hence $(T, \leq)$ is a Tree Structure representing the management hierarchy.

Example 4.5 (Biological Taxonomic Tree)**.** Fix one kingdom, say Animalia, and let

$$T = \{\text{Animalia, Chordata, Mammalia, Primates, Hominidae, Homo, Homo sapiens}\}.$$

Define

$$t_1 \ \leq \ t_2 \quad \iff \quad t_1 \text{ is a higher-rank taxon above } t_2.$$

Here "Animalia" is the unique minimal element; for each taxon t, its ancestral ranks form a totally ordered prefix chain; and $\leq$ is a partial order. Thus $(T, \leq)$ is a Tree Structure capturing the taxonomic classification down to species.

Example 4.6 (XML Document Object Model (DOM) Tree)**.** Let T be the set of all nodes in a well-formed XML document, including the root element, its child elements, text nodes, etc. Define

$$x \ \leq \ y \quad \iff \quad x \text{ is an ancestor of } y \text{ in the DOM tree (or } x = y\text{).}$$

The document's root element is the unique minimal element; each node's ancestor-chain is totally ordered by ancestry; and the relation is a partial order. Hence $(T, \leq)$ is a Tree Structure modeling the XML DOM hierarchy.

5 Forest Structure

A Forest Structure is a finite poset whose principal ideals are chains, decomposing into disjoint rooted trees across independent components.

Definition 5.1 (Forest Structure)**.** (cf. [65, 66]) Let F be a nonempty finite set, and let $\leq$ be a binary relation on F. We say that $(F, \leq)$ is a *Forest Structure* if and only if the following conditions hold:

(i) $\leq$ is a partial order on F. In other words, for all $x, y, z \in F$:

- $x \leq x$ (reflexivity),
- If $x \leq y$ and $y \leq x$, then $x = y$ (antisymmetry),
- If $x \leq y$ and $y \leq z$, then $x \leq z$ (transitivity).

(ii) For each $x \in F$, the *principal ideal*

$$\downarrow x \ := \ \{\, y \in F \mid y \leq x \,\}$$

is totally ordered by $\leq$. That is, if $y_1, y_2 \in \downarrow x$, then either $y_1 \leq y_2$ or $y_2 \leq y_1$.

Equivalently, $(F, \leq)$ is a finite poset whose Hasse diagram is a disjoint union of rooted trees; each connected component is a rooted tree. We call $(F, \leq)$ a *Forest Structure*.

Definition 5.2 (Forest Operation)**.** (cf. [65, 66]) Let $(F, \leq)$ be a Forest Structure as above. A *Forest Operation* of arity m (for some integer $m \geq 1$) is any function

$$\star : F^m \ \longrightarrow \ F.$$

In other words, a Forest Operation is an m-ary mapping that assigns to each m-tuple of elements of F another element of F.

Optionally, one may impose additional properties on $\star$, for instance:

- **Monotonicity:** $\star$ is *monotone* (order-preserving) if for all

$$(x_1, x_2, \ldots, x_m),\ (y_1, y_2, \ldots, y_m) \in F^m,$$

whenever $x_i \le y_i$ in F for each $i = 1, 2, \ldots, m$, then

$$\star(x_1,\ x_2,\ \ldots,\ x_m)\ \le\ \star(y_1,\ y_2,\ \ldots,\ y_m) \quad \text{in } F.$$

- As usual, when $m = 1$ one calls $\star$ a *unary forest operation*, and for $m = 2$ a *binary forest operation*, etc.

If no further axioms are imposed, any such mapping qualifies as a Forest Operation of arity m.

Example 5.3 (File System Directory Forest)**.** Let

$$F = \{\texttt{C:}, \texttt{C:\textbackslash Folder}, \texttt{C:\textbackslash Folder\textbackslash File.txt}, \texttt{D:}, \ldots\}$$

be the set of all file and folder paths on a computer with multiple drives (`C:`, `D:`, etc.). Define a partial order $\le$ by "is a prefix of":

$$p \ \le\ q \quad \Longleftrightarrow \quad p \text{ is a prefix path of } q.$$

Then each drive root (e.g. `C:`, `D:`) is a minimal element, and for any path q, the set $\{\, p \mid p \le q\}$ is totally ordered by prefix inclusion. Hence $(F, \le)$ is a Forest Structure whose connected components are the directory-trees on each drive.

Example 5.4 (Corporate Division Forest)**.** Let

$$F = \{\text{Masafumi, Yutaka, Tae, Dave, Eve, } \ldots\}$$

be employees across two independent divisions $\mathcal{D}_1$ and $\mathcal{D}_2$. Define $\le$ by "reports-to":

$$x \ \le\ y \quad \Longleftrightarrow \quad x \text{ is on the chain of command from } y \text{ up to a division head.}$$

Each division head (e.g. Tae in $\mathcal{D}_1$, Eve in $\mathcal{D}_2$) is minimal in its component, and any employee's chain to the head is totally ordered. Thus $(F, \le)$ is a Forest Structure comprising two disjoint organizational trees.

Example 5.5 (Biological Taxonomy Forest)**.** Let

$$F = \{\text{Kingdom, Phylum, Class, Order,}$$

$$\text{Family, Genus, Species, } \ldots\}$$

be taxa across multiple kingdoms (e.g. Animalia, Plantae). Define $\le$ by "is an ancestor taxon of":

$$t_1 \ \le\ t_2 \quad \Longleftrightarrow \quad t_1 \text{ is a higher-rank taxon containing } t_2.$$

Each kingdom is minimal in its component, and for any taxon t, the set of its ancestral ranks is totally ordered. Therefore $(F, \le)$ is a Forest Structure whose components are the separate taxonomic hierarchies.

Theorem 5.6 (Forest Structures Generalize Trees and Classical Sets)**.**

(a) *Every Tree Structure is a Forest Structure with exactly one connected component.*

(b) *Every set H equipped with the trivial order ($x \le y$ iff $x = y$) is both a Tree Structure (single-node trees) and hence a Forest Structure, recovering the bare Classical Structure on H.*

(c) *Conversely, any Forest Structure $(F, \le)$ decomposes uniquely as the disjoint union of its tree-components: each connected component of the Hasse diagram is a Tree Structure on its subset of F.*

In particular, the notion of Forest Structure strictly generalizes both Tree Structures (case of one component) and Classical Structures (case of trivial order).

Proof. **(a) Tree $\subseteq$ Forest.** Let $(T, \le)$ be a Tree Structure. Since a tree is by definition a connected rooted poset whose principal ideals are chains, it trivially satisfies the two axioms of a Forest Structure:

- $\leq$ is a partial order on T.
- For each $x \in T$, its principal ideal $\downarrow x$ is a chain.

Moreover, its Hasse diagram is a single rooted tree, so it is a Forest with one component.

(b) Trivial order $\Rightarrow$ Classical. Let H be any nonempty set, and define $x \leq y$ iff $x = y$. Then:

- Axiom (i) holds (reflexivity, antisymmetry, transitivity are trivial).
- For each x, $\downarrow x = \{x\}$ is vacuously totally ordered.

Thus $(H, \leq)$ is a Forest Structure whose Hasse diagram is a disjoint union of singleton trees. Forgetting the order recovers the Classical Structure (H).

(c) Forest $\rightarrow$ disjoint trees. Given a Forest Structure $(F, \leq)$, its Hasse diagram is a disjoint union of rooted trees by definition. Let $F = \bigsqcup_{i \in I} T_i$ be the partition into connected components. Restricting $\leq$ to each T_i yields a partial order with a unique minimal root and chain-like principal ideals, hence a Tree Structure on T_i.

Thus every Forest Structure breaks into Tree Structures, and both Trees and trivial posets embed as special cases. This shows that Forest Structures strictly generalize Tree and Classical Structures. □

6 Multi-Structure

A Multi-Structure replaces classical operations with maps from tuples to finite multisets, enabling multiple outputs per input tuple flexibly simultaneously.

Definition 6.1 (Finite Multiset)**.** (cf. [67–70]) Let H be a nonempty set. A *finite multiset* on H is a function

$$m : H \longrightarrow \mathbb{N}_0$$

with finite support $\{x \in H \mid m(x) > 0\}$. We denote by $\mathcal{M}(H)$ the collection of all such finite multisets on H. Equivalently, an element of $\mathcal{M}(H)$ can be written as $\{x_1^{k_1}, x_2^{k_2}, \dots, x_r^{k_r}\}$, where each $x_i \in H$ and $k_i = m(x_i) \in \mathbb{N}$.

Definition 6.2 (MultiOperation)**.** Let H be a nonempty set and fix an integer $m \geq 1$. A *multi-operation* of arity m on H is a map

$$\#^{(m)} : H^m \longrightarrow \mathcal{M}(H),$$
$$(x_1, \dots, x_m) \mapsto \#^{(m)}(x_1, \dots, x_m) \in \mathcal{M}(H).$$

Thus, instead of producing a single element of H, a multi-operation assigns a finite multiset of elements of H.

Definition 6.3 (MultiStructure)**.** A *MultiStructure* is a pair

$$\mathcal{MS} = \big(H, \{\#^{(m)} : H^m \to \mathcal{M}(H)\}_{m \in \mathcal{I}}\big),$$

where H is a nonempty carrier set and $\mathcal{I} \subseteq \mathbb{Z}_{>0}$ indexes a family of multi-operations of various arities. No further axioms are imposed unless specified.

Example 6.4 (Chemical Reaction MultiStructure)**.** Let

$$H = \{H_2, O_2, H_2O, CO_2, \dots\}$$

be the set of chemical species. A binary multi-operation

$$\#^{(2)} : H^2 \longrightarrow \mathcal{M}(H)$$

can model stoichiometric reactions. For instance,

$$\#^{(2)}(H_2, O_2) = \{H_2O^2\},$$

meaning one H_2 molecule and one O_2 molecule yield two water molecules. Similarly,

$$\#^{(2)}(C, O_2) = \{CO_2\},$$

models carbon combustion. Thus $\mathcal{MS} = (H, \{\#^{(2)}\})$ captures chemical reaction networks as a MultiStructure.

Example 6.5 (Recipe Composition MultiStructure)**.** Let

$$H = \{\text{flour, egg, sugar, butter, } \dots\}$$

be basic ingredients. A ternary multi-operation

$$\#^{(3)} : H^3 \longrightarrow \mathcal{M}(H)$$

forms a baking mixture. For example,

$$\#^{(3)}(\text{flour, egg, sugar}) = \{\text{flour, egg}^2\text{, sugar}\},$$

indicating the recipe uses one cup of flour, two eggs, and one cup sugar. The result is a multiset of ingredients for the cake batter. Here $\mathcal{MS}$ models culinary formulations as a MultiStructure.

Example 6.6 (Team Collaboration MultiStructure)**.** Let

$$H = \{\text{Masafumi, Yutaka, Tae, Dave}\}$$

be employees. A quaternary multi-operation

$$\#^{(4)} : H^4 \longrightarrow \mathcal{M}(H)$$

assigns member weights in a project. E.g.,

$$\#^{(4)}(\text{Masafumi, Yutaka, Yutaka, Tae}) = \{\text{Masafumi, Yutaka}^2\text{, Tae}\},$$

designates a four-person taskforce where Yutaka has two sub-roles. This $\#^{(4)}$ encodes team composition with multiplicities, making $\mathcal{MS} = (H, \{\#^{(4)}\})$ a MultiStructure for project staffing.

Theorem 6.7 (MultiStructure Generalizes Classical Structure)**.** *Every Classical Structure* $\big(H, \{\star^{(m)} : H^m \to H\}_{m\in\mathcal{I}}\big)$ *arises as a special case of a MultiStructure by identifying each operation with the corresponding* singleton *multiset operation. Conversely, any MultiStructure whose multi-operations always produce singleton multisets induces a classical structure.*

Proof. Let $\big(H, \{\star^{(m)}\}\big)$ be a Classical Structure. Define for each $m \in \mathcal{I}$ a multi-operation

$$\#^{(m)}(x_1, \dots, x_m) \;=\; \big\{\star^{(m)}(x_1, \dots, x_m)\big\} \;\in\; \mathcal{M}(H), \quad x_1, \dots, x_m \in H,$$

the multiset containing exactly one copy of the classical product. Clearly $\#^{(m)} : H^m \to \mathcal{M}(H)$ and the resulting $\mathcal{MS} = (H, \{\#^{(m)}\})$ is a MultiStructure.

Conversely, suppose $\mathcal{MS} = (H, \{\#^{(m)}\})$ is a MultiStructure in which

$$\#^{(m)}(x_1, \dots, x_m) \;=\; \{\,y\} \quad \text{(a singleton multiset)}$$

for every $(x_1, \dots, x_m) \in H^m$. Then the assignment $\star^{(m)}(x_1, \dots, x_m) = y$ is well-defined and yields an m-ary classical operation $\star^{(m)} : H^m \to H$. The family $\{\star^{(m)}\}$ satisfies exactly the axioms of the original MultiStructure restricted to singletons, and thus defines a Classical Structure.

Therefore MultiStructures strictly generalize Classical Structures. □

7 Iterative Multi-Structure

An Iterative Multi-Structure extends multiset operations across levels, combining multisets of multisets iteratively through k hierarchical stages in layered aggregation.

Definition 7.1 (Iterative Multi-Structure of Order k)**.** Let H be a nonempty set and fix an integer $k \geq 1$. Define iteratively the *multiset powersets*

$$\mathcal{M}^0(H) \;=\; H, \quad \mathcal{M}^{i+1}(H) \;=\; \mathcal{M}\big(\mathcal{M}^i(H)\big), \quad i = 0, 1, \ldots, k-1,$$

where $\mathcal{M}(X)$ denotes the collection of finite multisets on X (Definition 6.1). Let $\mathcal{I} \subseteq \mathbb{Z}_{>0}$ index a family of arities. An *Iterative Multi-Structure of order* k is a tuple

$$\mathcal{IMS}^{(k)} \;=\; \Big(H,\; \big\{\#^{(m,i)} : \big(\mathcal{M}^i(H)\big)^m \;\longrightarrow\; \mathcal{M}^{i+1}(H)\big\}_{m \in \mathcal{I},\, 0 \leq i < k}\Big),$$

where for each $i = 0, \ldots, k-1$ and each $m \in \mathcal{I}$,

$$\#^{(m,i)}(x_1, \ldots, x_m) \;\in\; \mathcal{M}^{i+1}(H), \quad x_j \in \mathcal{M}^i(H).$$

Thus $\#^{(m,0)}$ is an ordinary Multi-Structure operation on H, $\#^{(m,1)}$ combines multisets of multisets, and so on, up to level k.

Example 7.2 (Two-Level Sales Aggregation ($k = 2$))**.** Let $H = \{\text{Laptop}, \text{Mouse}, \text{Keyboard}\}$ be a set of products. Then

$$\mathcal{M}^0(H) = H, \quad \mathcal{M}^1(H) = \{\text{finite multisets of products}\}, \quad \mathcal{M}^2(H) = \{\text{finite multisets of product-multisets}\}.$$

Define two operations of arity $m = 2$:

$$\#^{(2,0)} : H^2 \to \mathcal{M}^1(H), \quad \#^{(2,0)}(x, y) = [\,x, y\,],$$

which bundles two products into a single sale event, and

$$\#^{(2,1)} : (\mathcal{M}^1(H))^2 \to \mathcal{M}^2(H), \quad \#^{(2,1)}(M_1, M_2) = [\,M_1, M_2\,],$$

which collects two sale events into a daily sales log. Thus $\mathcal{IMS}^{(2)} = (H, \{\#^{(2,0)}, \#^{(2,1)}\})$ models both individual sales and their aggregation into daily logs.

Example 7.3 (Three-Level Sensor Readings ($k = 3$))**.** Let $H = \{\text{Temp}, \text{Humid}\}$ be sensor types. Then

$$\mathcal{M}^0(H) = H, \quad \mathcal{M}^1(H) = \{\text{multisets of readings}\}, \quad \mathcal{M}^2(H) = \{\text{multisets of multisets (hourly logs)}\},$$

$$\mathcal{M}^3(H) = \{\text{multisets of hourly-log multisets (daily logs)}\}.$$

Define:

$$\#^{(1,0)} : H \to \mathcal{M}^1(H), \quad \#^{(1,0)}(\text{Temp}) = [\text{Temp}, \text{Temp}],$$

duplicating a reading for noise estimation;

$$\#^{(2,1)} : (\mathcal{M}^1(H))^2 \to \mathcal{M}^2(H), \quad \#^{(2,1)}(R_1, R_2) = [R_1, R_2],$$

combining two readings into an hourly log; and

$$\#^{(3,2)} : (\mathcal{M}^2(H))^3 \to \mathcal{M}^3(H), \quad \#^{(3,2)}(L_1, L_2, L_3) = [L_1, L_2, L_3],$$

assembling three hourly logs into a daily log. Hence $\mathcal{IMS}^{(3)} = (H, \{\#^{(1,0)}, \#^{(2,1)}, \#^{(3,2)}\})$ captures sensor readings at nested time scales.

Example 7.4 (Team-of-Teams Project Structure ($k = 2$))**.** Let $H = \{\text{A}, \text{B}, \text{C}\}$ be individual contributors. Then

$$\mathcal{M}^1(H) = \{\text{project teams}\}, \quad \mathcal{M}^2(H) = \{\text{collections of teams (departments)}\}.$$

Define:

$$\#^{(2,0)}(x, y) = \{x, y\}, \quad \#^{(2,1)}(T_1, T_2) = \{T_1, T_2\}.$$

Here $\#^{(2,0)}$ forms a two-person team, and $\#^{(2,1)}$ groups two teams into a department. Thus $\mathcal{IMS}^{(2)}$ models both team creation and departmental organization.

Theorem 7.5 (Iterative Multi-Structure Generalizes Multi- and Classical Structures)**.** *(1) When $k = 1$, $\mathcal{IMS}^{(1)}$ reduces exactly to the MultiStructure of Definition 6.3.*

(2) If in addition each multi-operation $\#^{(m,i)}$ always produces a singleton *multiset at every level i, then $\mathcal{IMS}^{(k)}$ induces a Classical Structure on H by collapsing each singleton to its sole element.*

Hence every MultiStructure and every Classical Structure arises as a specialization of an Iterative Multi-Structure.

Proof. **(1) Reduction to MultiStructure.** By definition $\mathcal{M}^0(H) = H$ and $\mathcal{M}^1(H) = \mathcal{M}(H)$. An Iterative Multi-Structure of order 1 consists exactly of the maps $\#^{(m,0)} : H^m \to \mathcal{M}(H)$, which is the data of a MultiStructure (Definition 6.3).

(2) Collapse to Classical Structure. Suppose each $\#^{(m,i)}(x_1, \ldots, x_m) \in \mathcal{M}^{i+1}(H)$ is a singleton multiset $\{y\}$. Then for every i and every m, define a classical operation

$$\star^{(m,i)}(x_1, \ldots, x_m) = y.$$

In particular for $i = 0$, $\star^{(m,0)}$ is an m-ary operation on H. The higher-level maps $\star^{(m,i)}$ act on iterated images of H, but the level-0 operation recovers a complete classical algebraic structure on the carrier H.

Thus by choosing $k = 1$ and/or by enforcing singleton outputs, one recovers exactly the MultiStructure or the Classical Structure, proving that $\mathcal{IMS}^{(k)}$ strictly generalizes both. □

8 U-(any-)Structure

Any-Structure is a multi-level algebraic framework where each level applies a chosen endofunctor (e.g. powerset, multiset, tree) and operations connect successive levels. And an U-Structure is a classical structure enriched by an uncertainty model U, assigning each element a membership degree in $\mathrm{Dom}(U)$ alongside its original operations.

Definition 8.1 (Any-Structure)**.** Let C be a category and let

$$\{\mathcal{G}_j : C \to \mathbf{Set}\}_{j \in J}$$

be a fixed collection of covariant endofunctors—each $\mathcal{G}_j$ being one of the constructors $\mathrm{Id},\ \mathcal{P},\ \mathrm{MSet},\ \mathrm{Tree},\ \mathrm{Forest},\ \ldots$. Choose a finite sequence of indices $(j_1, j_2, \ldots, j_k)$ in J, and define inductively

$$F_0 = \mathrm{Id}, \qquad F_i = \mathcal{G}_{j_i} \circ F_{i-1} \quad (1 \le i \le k).$$

An *Any-Structure of signature* $(j_1, \ldots, j_k)$ on an object $H \in \mathrm{Ob}(C)$ is the data

$$\mathcal{A} = \Big(H,\ \big\{\#^{(m,i)} : F_i(H)^m \longrightarrow F_{i+1}(H)\big\}_{1 \le i \le k,\ m \in I_i}\Big),$$

where each $\#^{(m,i)}$ is an m-ary operation at "level" i. In particular:

- $\#^{(m,1)}$ combines m-tuples in $H^m = F_0(H)^m$ to multisets or subsets in $F_1(H)$.
- $\#^{(m,2)}$ combines m-tuples in $F_1(H)^m$ to elements of $F_2(H)$, and so on.

Example 8.2 (Simple Graph as an Any-Structure)**.** Let $C = \mathbf{Set}$ and choose the signature $(\mathcal{P})$, so

$$F_0 = \mathrm{Id}, \quad F_1 = \mathcal{P}.$$

Take a vertex set H. Then

$$\#^{(2,1)} : H^2 \longrightarrow \mathcal{P}(H \times H), \quad \#^{(2,1)}(u, v) = \{(u, v)\}$$

defines a simple (directed) graph: each ordered pair (u, v) is sent to the singleton edge set $\{(u, v)\}$. No higher levels appear. Thus $\big(H, \{\#^{(2,1)}\}\big)$ is the classical directed-graph Any-Structure.

Example 8.3 (Multigraph as an Any-Structure)**.** Let C = **Set** and signature (MSet), so

$$F_0 = \mathrm{Id}, \quad F_1 = \mathrm{MSet}.$$

For a vertex set H, define

$$\#^{(2,1)} : H^2 \longrightarrow \mathrm{MSet}(H \times H), \quad \#^{(2,1)}(u, v) = [\,(u, v),\ (u, v)\,]$$

(the multiset with two parallel edges from u to v). More generally one could choose any multiplicity. This yields a two-level Any-Structure encoding a multigraph.

Theorem 8.4 (Recovery of All Previous Structures)**.** *By appropriate choice of* $(j_1, \ldots, j_k)$ *and arities* $\mathcal{I}_i$*, every structure studied above arises as a special case of an Any-Structure:*

(1) HyperStructure: $k = 1$, $j_1 = \mathcal{P}$, $\mathcal{I}_1 = \{2\}$.

(2) SuperHyperStructure: $k = 2$, $j_1 = j_2 = \mathcal{P}$, $\mathcal{I}_1 = \mathcal{I}_2 = \{2\}$.

(3) MultiStructure of arity m: $k = 1$, $j_1 = \mathrm{MSet}$, $\mathcal{I}_1 = \{m\}$.

(4) Iterative MultiStructure of order k: $j_1 = \cdots = j_k = \mathrm{MSet}$, *each* $\mathcal{I}_i = \{m_i\}$.

(5) TreeStructure: $k = 1$, $j_1 = \mathrm{Tree}$, $\mathcal{I}_1$ *arbitrary.*

(6) ForestStructure: $k = 1$, $j_1 = \mathrm{Forest}$, $\mathcal{I}_1$ *arbitrary.*

Proof. In each case one sets the sequence of functors F_i precisely to reproduce the level-i carrier of the target structure, and chooses the operation $\#^{(m,i)}$ to match its defining m-ary map. For example, for SuperHyper-Structure we take

$$F_0 = \mathrm{Id}, \quad F_1 = \mathcal{P}, \quad F_2 = \mathcal{P} \circ \mathcal{P},$$

and let $\#^{(2,1)} : H^2 \to \mathcal{P}(H)$ be the hyperoperation, $\#^{(2,2)} : \mathcal{P}(H)^2 \to \mathcal{P}^2(H)$ the superhyperoperation. The other instances follow analogously, verifying that Any-Structure indeed subsumes all previously defined structures. □

Definition 8.5 (Uncertain Model)**.** (cf. [71]) Let $\mathbb{U}$ denote the collection of all *Uncertain* models, where each $U \in \mathbb{U}$ is characterized by a mapping

$$\mu_U : X_U \longrightarrow \mathrm{Dom}(U),$$

from some universe of discourse X_U to a set of admissible *membership-degree tuples* $\mathrm{Dom}(U) \subseteq [0, 1]^r$ (for an appropriate integer $r \geq 1$), subject to specific algebraic or geometric constraints depending on U. In particular, we include at least the following types:

1. **Fuzzy [72–74].**
$$\mathrm{Dom}(\mathrm{Fuzzy}) = [0, 1].$$

2. **Intuitionistic Fuzzy [75–78].**
$$\mathrm{Dom}\big(\text{Intuitionistic Fuzzy}\big) = \big\{(\mu, \nu) \in [0, 1]^2 \;\big|\; \mu + \nu \leq 1\big\}.$$

3. **Inconsistent Intuitionistic Fuzzy (Picture Fuzzy, Ternary Fuzzy) [79,80].**
$$\mathrm{Dom}\big(\text{Inconsistent Intuitionistic Fuzzy}\big) =$$
$$\big\{(\mu, \nu, \pi) \in [0, 1]^3 \;\big|\; \mu + \nu + \pi \leq 2\big\},$$
where μ, ν, and π represent membership, non-membership, and refusal degrees respectively.

4. **Pythagorean Fuzzy [81,82].**
$$\mathrm{Dom}\big(\text{Pythagorean Fuzzy}\big) =$$
$$\big\{(\mu, \nu) \in [0, 1]^2 \;\big|\; \mu^2 + \nu^2 \leq 1\big\}.$$

5. **q-Rung Orthopair Fuzzy [83,84].**

$$\mathrm{Dom}\big(\text{q-Rung Orthopair Fuzzy}\big) \;=\; \big\{(\mu,\nu)\in[0,1]^2 \;\big|\; \mu^q+\nu^q\le 1\big\},\quad q\ge 1.$$

6. **Neutrosophic [85–90].**

$$\mathrm{Dom}\big(\text{Neutrosophic}\big) \;=\; \big\{(T,I,F)\in[0,1]^3 \;\big|\; 0\le T+I+F\le 3\big\}.$$

7. **Refined Neutrosophic [91–93].**

$$\mathrm{Dom}\big(\text{Refined Neutrosophic}\big) \;=\; \Big\{\big(T_1,T_2,\dots,T_k;\ I_1,I_2,\dots,I_k;\ F_1,F_2,\dots,F_k\big)\in[0,1]^{3k} \;\Big|\; \sum_{j=1}^{k}(T_j+I_j+F_j)\le 3k\Big\}.$$

8. **Plithogenic [94–98].** Let P be a set of primary attributes and P_v the set of their possible values. Then

$$\mathrm{Dom}\big(\text{Plithogenic}\big) \;=\; \left\{\big(x,\ v,\ \mathrm{pdf}(x,v),\ \mathrm{pCF}(v_1,v_2)\big) \;\middle|\; \begin{array}{l} x\in P,\ v\in P_v,\\ \mathrm{pdf}(x,v)\in[0,1]^s,\ \mathrm{pCF}(v_1,v_2)\in[0,1]^t \end{array}\right\}.$$

9. … Additional uncertainty models may be included, each with its own domain $\mathrm{Dom}(U)\subseteq[0,1]^r$ defined by suitable constraints.

We refer to any element of $\mathrm{Dom}(U)$ as a *U-membership degree*.

Let $U\in\mathbb{U}$ be a fixed uncertainty model with degree-domain $\mathrm{Dom}(U)$. Recall that in this paper, the term *Structure* refers to an arbitrary mathematical structure, including but not limited to those from the domains of Set Theory, Logic, Probability, Statistics, Algebra, and Geometry.

Definition 8.6 (U-Structure)**.** Let $C=(H,\{\#^{(m)}\})$ be a Classical Structure as in Definition 1.1, with carrier H. A *U-Structure C^U* is the pair

$$C^U \;=\; \Big(H,\ \mu\colon H\ \longrightarrow\ \mathrm{Dom}(U),\ \{\#^{(m)}\}\Big),$$

where

$$\mu(x)\in\mathrm{Dom}(U)\quad\text{is the } U\text{-membership degree of } x\in H.$$

The underlying classical operations $\{\#^{(m)}\}$ remain those of C, but each element of H carries a U-membership label $\mu(x)$. In effect, C^U is a *fuzzy- or neutrosophic*-enrichment of the classical structure C.

Example 8.7 (Fuzzy Set as a Fuzzy-U-Structure)**.** Let $H=\{a,b,c\}$ be a simple carrier with no additional operations, and let

$$U=\text{Fuzzy},\qquad \mathrm{Dom}(U)=[0,1].$$

Define

$$\mu\colon H\to[0,1],\quad \mu(a)=0.9,\ \mu(b)=0.6,\ \mu(c)=0.3.$$

Then

$$C^U=\big(H,\ \mu,\ \text{(no operations)}\big)$$

is a Fuzzy-U-Structure: each element carries a fuzzy membership degree in $[0,1]$.

Example 8.8 (Fuzzy Graph as a Fuzzy-U-Structure)**.** (cf. [99–102]) Let

$$H=\{1,2,3\},\quad \#^{(2)}(i,j)=\begin{cases}\{\,\{i,j\}\} & \text{if } |i-j|=1,\\ \varnothing & \text{otherwise,}\end{cases}$$

be the underlying simple graph hyperoperation, and let

$$U=\text{Fuzzy},\qquad \mathrm{Dom}(U)=[0,1].$$

Assign fuzzy-membership to vertices by

$$\mu(1) = 1.0, \quad \mu(2) = 0.8, \quad \mu(3) = 0.5.$$

Then

$$C^U = \Big(H,\ \mu\colon H \to [0,1],\ \#^{(2)}\Big)$$

is a Fuzzy Graph-U-Structure: vertices carry fuzzy grades, edges follow the classical adjacency rule.

Example 8.9 (Neutrosophic Group as a Neutrosophic-U-Structure)**.** (cf. [43, 103]) Let

$$H = (\mathbb{Z}, +) \quad \text{(the integers under addition)},$$

and let

$$U = \text{Neutrosophic}, \quad \mathrm{Dom}(U) = \{(T, I, F) \in [0,1]^3 \mid 0 \le T + I + F \le 3\}.$$

Define a neutrosophic membership map reflecting uncertainty in group identity:

$$\mu(n) = \big(T_n, I_n, F_n\big), \quad T_n = e^{-|n|},\ I_n = 0.1,\ F_n = 1 - T_n - I_n.$$

Then

$$C^U = \big(\mathbb{Z},\ \mu\colon \mathbb{Z} \to \mathrm{Dom}(U),\ +\big)$$

is a Neutrosophic Group-U-Structure: each integer is labeled by a neutrosophic triple (T, I, F).

An U-any-Structure is an any-Structure augmented with a U-membership map on the base set, unifying levelled operations and fuzzy/neutrosophic enrichment in one framework.

Definition 8.10 (U-any-Structure)**.** Let C be a category, let $\{\mathcal{G}_j : C \to \mathbf{Set}\}_{j \in J}$ be a fixed family of endofunctors (e.g. Id, $\mathcal{P}$, MSet, Tree, Forest, . . .), and choose a finite signature $(j_1, \ldots, j_k)$ in J. Define

$$F_0 = \mathrm{Id}, \qquad F_i = \mathcal{G}_{j_i} \circ F_{i-1} \quad (1 \le i \le k).$$

Let $U \in \mathbb{U}$ be a fixed uncertainty model with degree-domain $\mathrm{Dom}(U) \subseteq [0,1]^r$. A *$U$-any-Structure of signature* $(j_1, \ldots, j_k)$ on an object $H \in \mathrm{Ob}(C)$ consists of:

$$\mathcal{A}^U \;=\; \Big(H,\ \mu\colon H \to \mathrm{Dom}(U),\ \{\#^{(m,i)} : F_i(H)^m \to F_{i+1}(H)\}_{1 \le i \le k,\ m \in I_i}\Big),$$

where

- $\mu(x) \in \mathrm{Dom}(U)$ assigns each base element $x \in H$ a U-membership degree.
- For each level i, $\#^{(m,i)}$ is an m-ary operation "at depth i"

$$\#^{(m,i)} \;:\; F_i(H)^m \;\longrightarrow\; F_{i+1}(H).$$

Example 8.11 (Fuzzy–Hyper-U-Any-Structure: Fuzzy Hypergraph)**.** Let $C = \mathbf{Set}$, $J = \{\mathcal{P}\}$, and choose signature (j_1) with $\mathcal{G}_{j_1} = \mathcal{P}$. Let $H = \{a, b, c\}$ and $U = $ Fuzzy with $\mathrm{Dom}(U) = [0,1]$. Then

$$F_0(H) = H, \qquad F_1(H) = \mathcal{P}(H).$$

Define

$$\mu\colon H \to [0,1], \quad \mu(a) = 0.9,\ \mu(b) = 0.7,\ \mu(c) = 0.5,$$

and the hyper-operation

$$\#^{(2,1)}\colon H^2 \to \mathcal{P}(H), \quad \#^{(2,1)}(x, y) = \{x, y\}.$$

Then $\mathcal{A}^U = (H, \mu, \{\#^{(2,1)}\})$ is a fuzzy–hyper-U-Any-Structure: each element carries a fuzzy degree, and any two elements form a hyperedge.

Example 8.12 (Neutrosophic–SuperHyper-U-Any-Structure: Neutrosophic Team Synergy)**.** Let C = **Set**, $J = \{\mathcal{P}, \mathcal{P}\}$, with signature (j_1, j_2) both equal to $\mathcal{P}$. Let $H = \{A, B\}$ and U = Neutrosophic with $\mathrm{Dom}(U) = \{(T, I, F) \in [0,1]^3 \mid T + I + F \leq 3\}$. Then

$$F_0(H) = H, \quad F_1(H) = \mathcal{P}(H), \quad F_2(H) = \mathcal{P}\big(\mathcal{P}(H)\big).$$

Define

$$\mu(A) = (0.8, 0.1, 0.1), \quad \mu(B) = (0.7, 0.2, 0.1),$$
$$\#^{(2,1)} : H^2 \to \mathcal{P}(H), \quad \#^{(2,1)}(x, y) = \{x, y\},$$
$$\#^{(2,2)} : \mathcal{P}(H)^2 \to \mathcal{P}^2(H), \quad \#^{(2,2)}(X, Y) = \{X, Y\}.$$

Then $\mathcal{A}^U = (H, \mu, \{\#^{(2,1)}, \#^{(2,2)}\})$ is a neutrosophic–superhyper-U-Any-Structure: each pair of individuals yields a team, and each pair of teams yields a "meta-team."

Example 8.13 (Fuzzy–Tree-U-Any-Structure: Fuzzy Decision Tree)**.** Let C be the category whose objects are finite rooted trees and J = {Tree} with signature (j_1), $\mathcal{G}_{j_1}$ = Tree. Fix $H = \{\mathrm{Yes}, \mathrm{No}\}$ as leaf set, and U = Fuzzy. Then

$$F_0(H) = H, \quad F_1(H) = \mathrm{Tree}(H)$$

is the set of all small decision-trees over $\{\mathrm{Yes}, \mathrm{No}\}$. Define

$$\mu(\mathrm{Yes}) = 0.95, \ \mu(\mathrm{No}) = 0.85,$$

and for each pair of trees $T_1, T_2 \in \mathrm{Tree}(H)$ choose a grafting operation

$$\#^{(2,1)} : \mathrm{Tree}(H)^2 \to \mathrm{Tree}(H), \quad \#^{(2,1)}(T_1, T_2) = \text{attach root of } T_2 \text{ as child of root of } T_1.$$

Then $\mathcal{A}^U = (H, \mu, \{\#^{(2,1)}\})$ is a fuzzy–tree-U-Any-Structure: each decision outcome has a fuzzy confidence, and any two trees can be combined into a larger decision-tree.

Theorem 8.14 (Generalization of U- and Any-Structures)**.** *Every U-Structure and every Any-Structure embeds into the U-any-Structure framework:*

(a) If $k = 0$ then $F_0 = \mathrm{Id}$, and Definition 8.10 reduces to $\big(H, \ \mu : H \to \mathrm{Dom}(U)\big)$, i.e. an ordinary U-Structure.

(b) If one forgets *the map $\mu : H \to \mathrm{Dom}(U)$, then $\mathcal{A}^U$ coincides with the Any-Structure of Definition 8.1 with signature $(j_1, \ldots, j_k)$.*

Hence the notion of U-any-Structure strictly unifies both U-Structures and Any-Structures.

Proof. **(a) Reduction to U-Structure.** When $k = 0$ there are no operations $\#^{(m,i)}$. We have $F_0 = \mathrm{Id}$ and the only data is $\mu : H \to \mathrm{Dom}(U)$, which is exactly a U-Structure on H (Definition 8.6).

(b) Reduction to Any-Structure. If we omit the membership map μ, the remaining tuple

$$\big(H, \ \{\#^{(m,i)}\}\big)$$

is precisely an Any-Structure of signature $(j_1, \ldots, j_k)$ on H (Definition 8.1).

Since neither reduction loses any required axioms, every U-Structure and every Any-Structure arises by specialization of the U-any-Structure. Conversely, any U-any-Structure contains both features simultaneously, proving the claim. □

9 Functorial-(any-)Structure

A functorial Structure is a covariant functor

$$F\colon C \longrightarrow \mathbf{Set}$$

assigns to each object a set of "structures" and to each morphism a pushforward mapping, respecting identities and composition. A functorial–Any–Structure is a multi–level functorial framework

$$F_0 = \mathrm{Id},\ F_i = \mathcal{G}_{j_i} \circ F_{i-1},$$

enriched by U-labels on base elements, with level-wise m-ary operations $\#^{(m,i)}\colon F_i(H)^m \to F_{i+1}(H)$.

Definition 9.1 (Functorial Set)**.** (cf. [71]) Let C be a category and

$$F\colon C \ \longrightarrow\ \mathbf{Set}$$

be a (covariant) endofunctor. For any object $X \in \mathrm{Ob}(C)$, an *F-set over X* is an element

$$s \ \in\ F(X).$$

We denote the collection of all F-sets over X simply by $F(X)$. A morphism $f\colon X \to Y$ in C induces a *pushforward*

$$F(f)\colon F(X) \ \longrightarrow\ F(Y), \qquad s \ \mapsto\ F(f)(s).$$

Definition 9.2 (Functorial Structure)**.** (cf. [71]) Let C be a category. A *Functorial Structure* on C is simply a covariant functor

$$F\ :\ C\ \longrightarrow\ \mathbf{Set}.$$

For each object $X \in \mathrm{Ob}(C)$, an element

$$s\ \in\ F(X)$$

is called an *F-structure on X*. Every morphism $f\colon X \to Y$ in C induces a *pushforward*

$$F(f)\colon F(X) \ \longrightarrow\ F(Y), \qquad s \ \longmapsto\ F(f)(s),$$

and the usual functoriality conditions $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$ and $F(g \circ f) = F(g) \circ F(f)$ hold.

Example 9.3 (Graph Structures)**.** Let $C = \mathbf{Set}$ and define the *graph-functor*

$$\mathsf{Graph}\colon \mathbf{Set} \longrightarrow \mathbf{Set}, \quad \mathsf{Graph}(X) = \{\, E \subseteq X \times X\},$$

sending a set X to the collection of all binary relations on X. For a function $f\colon X \to Y$,

$$\mathsf{Graph}(f)\colon \mathsf{Graph}(X) \to \mathsf{Graph}(Y), \quad E \mapsto \{(f(x), f(x')) \mid (x, x') \in E\}.$$

An element $E \in \mathsf{Graph}(X)$ is precisely a *graph structure* on X, and $\mathsf{Graph}(f)$ pushes edges forward along f.

Example 9.4 (Topological Structures)**.** Let $C = \mathbf{Set}$ and set

$$\mathsf{Top}\colon \mathbf{Set} \longrightarrow \mathbf{Set}, \quad \mathsf{Top}(X) = \big\{\tau \subseteq \mathcal{P}(X) \bigm| \tau \text{ is a topology on } X\big\}.$$

For $f\colon X \to Y$, define

$$\mathsf{Top}(f)\colon \mathsf{Top}(X) \to \mathsf{Top}(Y), \quad \tau \mapsto \{\, B \subseteq Y \mid f^{-1}(B) \in \tau\}.$$

Each $\tau \in \mathsf{Top}(X)$ is a *topological structure* on X, and $\mathsf{Top}(f)$ is the usual inverse-image map carrying open sets forward.

Example 9.5 (Vector Bundle Structures)**.** Let $C = \mathbf{Top}$ be the category of topological spaces and continuous maps. Define

$$\mathsf{VB}\colon \mathbf{Top} \longrightarrow \mathbf{Set}, \quad \mathsf{VB}(X) = \big\{\pi\colon E \to X \bigm| E \text{ is a rank-}n\text{ vector bundle over } X\big\}.$$

For a continuous map $f\colon X \to Y$, the pullback bundle functor

$$\mathsf{VB}(f)\colon \mathsf{VB}(Y) \to \mathsf{VB}(X), \quad \pi \mapsto f^*\pi$$

assigns to each bundle $E \to Y$ its pullback $f^*E \to X$. Thus an element of $\mathsf{VB}(X)$ is a *vector bundle structure* on X.

Example 9.6 (Group Object Structures)**.** Let C be any category with finite products. The functor

$$\mathsf{GrpObj}\colon C \longrightarrow \mathbf{Set}, \quad \mathsf{GrpObj}(X) = \{ (\mu, \eta, i) \mid X \text{ equipped with multiplication, unit, inverse in } C \}$$

assigns to each object X the set of all possible group-object structures on X. For a morphism $f\colon X \to Y$, $\mathsf{GrpObj}(f)$ carries one group-object structure on X to another on Y via the universal property of products and pullbacks.

Definition 9.7 (Functorial-Any-Structure)**.** Let C be a category, let $\{\mathcal{G}_j : C \to \mathbf{Set}\}_{j\in J}$ be a fixed family of covariant endofunctors (e.g. $\mathrm{Id}, \mathcal{P}, \mathrm{MSet}, \mathrm{Tree}, \mathrm{Forest}, \ldots$), and choose a finite *signature* $(j_1, \ldots, j_k)$ in J. Define functors

$$F_0 = \mathrm{Id}, \qquad F_i = \mathcal{G}_{j_i} \circ F_{i-1} \quad (1 \le i \le k).$$

Fix an uncertainty model U from a class $\mathbb{U}$ with degree-domain $\mathrm{Dom}(U) \subseteq [0,1]^r$. A *Functorial-Any-Structure of signature* $(j_1, \ldots, j_k)$ *enriched by* U on an object $H \in \mathrm{Ob}(C)$ consists of the data

$$\mathcal{FAS}^U = \Big(H, \underbrace{\mu\colon H \longrightarrow \mathrm{Dom}(U)}_{U\text{-label}}, \{\#^{(m,i)} : F_i(H)^m \to F_{i+1}(H)\}_{1\le i\le k,\, m\in\mathcal{I}_i}\Big),$$

where

- $\mu(x) \in \mathrm{Dom}(U)$ assigns to each $x \in H$ a U-membership degree.
- For each level $i = 0, \ldots, k-1$ and each arity $m \in \mathcal{I}_i$,

  $$\#^{(m,i)} : F_i(H)^m \longrightarrow F_{i+1}(H)$$

  is an m-ary operation "at depth i."

Functoriality of each $\mathcal{G}_{j_i}$ provides canonical "pushforwards" on the carriers $F_i(H)$.

Example 9.8 (Fuzzy Hypergraph as a Functorial-Any-Structure)**.** Let $C = \mathbf{Set}$, $J = \{\mathcal{P}\}$, and take the signature $(\mathcal{P})$ of length $k = 1$. Thus

$$F_0 = \mathrm{Id}, \quad F_1 = \mathcal{P} \circ \mathrm{Id} = \mathcal{P}.$$

Fix the fuzzy-membership model U with $\mathrm{Dom}(U) = [0,1]$. For a finite vertex set H, define

$$\mu\colon H \longrightarrow [0,1],$$

assigning each vertex a membership degree. The single operation

$$\#^{(2,0)} : H \times H \longrightarrow \mathcal{P}(H), \quad (u,v) \mapsto \{u,v\},$$

creates an (ordinary) hyperedge from any pair of vertices. The result

$$\mathcal{FAS}^U = (H, \mu, \#^{(2,0)})$$

is precisely a fuzzy-hypergraph: each vertex has degree $\mu(x) \in [0,1]$, and hyperedges are two-element subsets of H.

Example 9.9 (Neutrosophic Multigraph as a Functorial-Any-Structure)**.** Let $C = \mathbf{Set}$, $J = \{\mathrm{MSet}\}$, signature (MSet). Thus

$$F_0 = \mathrm{Id}, \quad F_1 = \mathrm{MSet} \circ \mathrm{Id} = \mathrm{MSet},$$

where $\mathrm{MSet}(H)$ is the set of finite multisets on H. Fix the neutrosophic model U with $\mathrm{Dom}(U) = \{(T, I, F) \in [0,1]^3 \mid 0 \le T + I + F \le 3\}$. Given H as the set of "nodes," let

$$\mu\colon H \longrightarrow \mathrm{Dom}(U),$$

label each node by truth/indeterminacy/falsity degrees. Define the parallel-edge operation

$$\#^{(2,0)} : H^2 \longrightarrow \mathrm{MSet}(H), \quad (u,v) \mapsto [\,u,v\,],$$

the multiset containing exactly one copy of u and one of v. Then

$$\mathcal{FAS}^U = (H, \mu, \#^{(2,0)})$$

is a neutrosophic multigraph: each node carries a neutrosophic label, and edges allow multiplicity.

Example 9.10 (Fuzzy Tree-of-Sets as a Functorial-Any-Structure)**.** Let C = **Set**, $J = \{\text{Tree}\}$, signature (Tree). So

$$F_0 = \text{Id}, \quad F_1 = \text{Tree} \circ \text{Id} = \text{Tree},$$

where $\text{Tree}(H)$ is the collection of all finite rooted trees whose leaves are labeled by elements of H. Choose the fuzzy model U with $\text{Dom}(U) = [0, 1]$. For a set of attributes H, let

$$\mu : H \longrightarrow [0, 1]$$

assign each attribute a confidence score. Define the trivial "singleton-tree" operation

$$\#^{(1,0)} \;:\; H \;\longrightarrow\; \text{Tree}(H), \quad x \;\mapsto\; \big(\bullet \xrightarrow{x}\big),$$

the one-leaf tree labeled by x. Then

$$\mathcal{FAS}^U = \big(H,\ \mu,\ \#^{(1,0)}\big)$$

is a fuzzy tree-of-sets: each attribute carries a fuzzy score, and each generates its own singleton tree.

Theorem 9.11 (Unification of Functorial, Any-, and U-Structures)**.** *The Functorial-Any-Structure of Definition 9.7 simultaneously generalizes:*

(a) (U-Structure*) If $k = 0$, then $F_0 = \text{Id}$ and no operations $\#^{(m,i)}$ are present. The data reduce to $\big(H,\ \mu : H \to \text{Dom}(U)\big)$, i.e. a U-Structure on H.*

(b) (Functorial Structure*) If one* forgets *μ and chooses each $\mathcal{I}_i$ and $\#^{(m,i)}$ arbitrarily, the remaining tuple $\big(H, \{\#^{(m,i)}\}\big)$ is exactly the Any-Structure of signature $(j_1, \ldots, j_k)$, which in turn is a Functorial Structure since each F_i is a composition of the endofunctors $\mathcal{G}_{j_i}$.*

(c) (Any-Structure*) If one also sets U to be the trivial one-point model* $\text{Dom}(U) = \{*\}$*, then μ is unique and can be ignored, recovering the pure Any-Structure.*

Proof. **(a) Reduction to U-Structure.** When $k = 0$ there are no operations $\#^{(m,i)}$. Since $F_0 = \text{Id}$, the only datum is $\mu : H \to \text{Dom}(U)$, which matches Definition 8.6 of a U-Structure.

(b) Reduction to Any-Structure. Omitting μ leaves $\big(H, \{\#^{(m,i)} : F_i(H)^m \to F_{i+1}(H)\}\big)$, which is precisely the Any-Structure of Definition 8.1 with signature $(j_1, \ldots, j_k)$. Moreover, since each F_i is a functor, this tuple is a Functorial Structure in the sense of Definition 9.2.

(c) Reduction to Functorial Structure. Finally, if we take U to be the trivial uncertainty model with $\text{Dom}(U) = \{*\}$, then μ carries no information and the Functorial-Any-Structure collapses to a Functorial Structure given by the composite functor $F_k = \mathcal{G}_{j_k} \circ \cdots \circ \mathcal{G}_{j_1}$.

Thus each special case is recovered by forgetting or trivializing the appropriate data, showing that the Functorial-Any-Structure indeed subsumes all three notions. □

10 Other Structure

The definitions of Curried Structure and Dynamic Structure are provided below.

10.1 Curried Structure

Currying transforms a k-ary operation

$$f : H^k \;\longrightarrow\; H$$

into a nested sequence of single-argument functions

$$f^c(x_1)(x_2)\cdots(x_k) \;=\; f(x_1, x_2, \ldots, x_k),$$

enabling elegant partial application and composition. This technique appears across many disciplines—from theoretical physics and programming-language semantics [104–106]—and underpins functional programming languages such as Haskell, C++, and others [107–111].

Definition 10.1 (Curried Function)**.** [112, 113] Given sets A, B, and C, a *curried function* is a map

$$f\colon A \longrightarrow (B \longrightarrow C),$$

so that each $a \in A$ determines a function $f(a)\colon B \to C$. Equivalently, currying establishes a bijection

$$\mathrm{curry}\colon (A \times B \to C) \longleftrightarrow \big(A \to (B \to C)\big).$$

Definition 10.2 (Curried k-ary Function)**.** (cf. [112, 113]) Let $A_1, A_2, \ldots, A_k, B$ be sets. A *curried k-ary function* is a map

$$f\colon A_1 \longrightarrow \big(A_2 \longrightarrow \cdots \longrightarrow (A_k \longrightarrow B)\cdots\big),$$

often abbreviated as

$$f\colon A_1 \to A_2 \to \cdots \to A_k \to B.$$

Example 10.3 (Shipping Cost Calculation as Curried Ternary Function)**.** In logistics, the shipping cost can be computed from three parameters: a per-unit rate, package weight, and travel distance. We model this as a curried 3-ary function

$$\mathit{shipCost}\colon \mathbb{R} \longrightarrow \mathbb{R} \longrightarrow \mathbb{R} \longrightarrow \mathbb{R},$$

defined by

$$\mathit{shipCost}(r)(w)(d) = r \times w \times d,$$

where

- r is the rate (e.g. dollars per kilogram-kilometer),
- w is the package weight in kilograms,
- d is the distance in kilometers,
- and the result is the total shipping cost in dollars.

This curried form allows partial application—for example, fixing a standard rate r_0 yields a simpler function

$$\mathit{shipCost}(r_0) : \mathbb{R} \to \mathbb{R} \to \mathbb{R},$$

which only requires weight and distance to compute cost under rate r_0.

Definition 10.4 (Curried Structure)**.** [112] Let H be a nonempty set. A *Curried Structure* on H is a pair

$$(H, \varphi), \quad \varphi\colon H \longrightarrow (H \longrightarrow H),$$

where $\varphi(x)$ is a function $H \to H$. Equivalently, $\varphi\colon H \to H \to H$.

Definition 10.5 (Curried k-ary Structure)**.** [112] Let H be a nonempty set (the *carrier*) and $\mathcal{I} \subseteq \mathbb{Z}_{>0}$ a family of arities. A *curried k-ary structure* of signature $\mathcal{I}$ on H consists of maps

$$\{\, f^{(m)} : H \to \underbrace{H \to \cdots \to H}_{m \text{ times}} \}_{m \in \mathcal{I}},$$

where each $f^{(m)}$ is termed the *curried m-ary operation*, sometimes notated

$$f^{(m)} : H \to H \to \cdots \to H \quad (m \text{ arrows}).$$

Example 10.6 (Sentence Construction as Curried Ternary Structure)**.** Let H be the set of all finite strings over a given alphabet, and let $\mathcal{I} = \{3\}$. Define the curried ternary operation

$$f^{(3)}\colon H \longrightarrow H \longrightarrow H \longrightarrow H$$

by

$$f^{(3)}(s_1)(s_2)(s_3) = s_1 \,\|\, \text{" "} \,\|\, s_2 \,\|\, \text{" "} \,\|\, s_3,$$

where $\|$ denotes string concatenation. For instance,

$$f^{(3)}(\text{"Alice"})(\text{"loves"})(\text{"Bob"}) = \text{"Alice loves Bob"}.$$

Partial application $f^{(3)}(\text{"Alice"})(\text{"loves"})\colon H \to H$ yields a binary function that completes sentences of the form “Alice loves . . . ”.

10.2 Dynamic Structure

A *Dynamic Structure* combines data and context through stateful operations. Each operation receives a current context and a fixed number of input values, produces an output value and an updated context, and thus models computations that evolve state over time. A *Dynamic HyperStructure* extends this idea to collections of values. Its operations accept a current context and several sets of elements, and produce a new set of elements together with an updated context, enabling dynamic aggregation and context tracking over groups of related items. A *Dynamic SuperHyperStructure* generalizes further to multi-level collections. Its operations take a current context and nested collections of elements, and return an updated nested collection and modified context, supporting hierarchical aggregation and stateful transformations across multiple layers.

Definition 10.7 (Dynamic Structure)**.** Let H be a nonempty set (the *carrier*) and let D be a nonempty set (the *dynamic context*). Fix a signature $\mathcal{I} \subseteq \mathbb{Z}_{>0}$ of arities. A *Dynamic Structure* of signature $\mathcal{I}$ on the pair (H, D) consists of a family of *dynamic m-ary operations*

$$\varphi^{(m)} : \ D \times H^m \ \longrightarrow \ H \times D, \qquad m \in \mathcal{I},$$

which update both an element of H and the context D. Equivalently, each $\varphi^{(m)}$ can be *curried* as

$$\varphi^{c,(m)} : \ D \ \longrightarrow \ H \ \longrightarrow \ \cdots \ \longrightarrow \ H \ \longrightarrow \ H \times D,$$

subject to

$$\varphi^{c,(m)}(d)\,(x_1)\cdots(x_m) \ = \ \varphi^{(m)}\big(d,\ (x_1, \ldots, x_m)\big).$$

Definition 10.8 (Dynamic HyperStructure)**.** Let H be a nonempty set and D a nonempty *dynamic context*. Fix a signature $\mathcal{I} \subseteq \mathbb{Z}_{>0}$. A *Dynamic HyperStructure* on (H, D) of signature $\mathcal{I}$ consists of, for each $m \in \mathcal{I}$, a map

$$\varphi^{(m)} : D \times \mathcal{P}(H)^m \ \longrightarrow \ \mathcal{P}(H) \times D,$$

or equivalently its *curried form*

$$\varphi^{c,(m)} : D \ \longrightarrow \ \underbrace{\mathcal{P}(H) \longrightarrow \cdots \longrightarrow \mathcal{P}(H)}_{m \text{ arrows}} \ \longrightarrow \ \mathcal{P}(H) \times D,$$

satisfying

$$\varphi^{c,(m)}(d)(A_1)\cdots(A_m) = \varphi^{(m)}\big(d,\ (A_1, \ldots, A_m)\big).$$

Example 10.9 (Warehouse Palletizing as a Dynamic HyperStructure)**.** Let H be the set of all package identifiers in a warehouse, and let D be the list of log entries recording packing actions. Fix arity $m = 2$ and define

$$\varphi^{(2)}\big(d, (B_1, B_2)\big) = \Big(B_1 \cup B_2,\ d \cup \{\text{“Merged boxes } B_1 \text{ and } B_2\text{”}\}\Big),$$

where $B_1, B_2 \subseteq H$ are two cartons. The operation produces a new pallet (the union of the cartons) and appends a descriptive entry to the log. This models dynamic subset aggregation with stateful context updates in a real-world logistics system.

Theorem 10.10 (Dynamic HyperStructure generalizes HyperStructure)**.** *Every ordinary HyperStructure*

$$\big(\mathcal{P}(H), \{\star^{(m)} : \mathcal{P}(H)^m \to \mathcal{P}(H)\}\big)$$

arises as the special case of a Dynamic HyperStructure by taking the trivial context $D = \{*\}$ and defining

$$\varphi^{(m)}(*, (A_1, \ldots, A_m)) = \big(\star^{(m)}(A_1, \ldots, A_m),\ *\big).$$

Conversely, any Dynamic HyperStructure on $(H, \{*\})$ reduces to an ordinary HyperStructure by projection to the first component.

Proof. Given a HyperStructure $(\mathcal{P}(H), \{\star^{(m)}\})$, set $D = \{*\}$ and define $\varphi^{(m)}(*, \vec{A}) = (\star^{(m)}(\vec{A}), *)$. Its curried form $\varphi^{c,(m)}(*)(A_1)\cdots(A_m) = (\star^{(m)}(A_1, \ldots, A_m), *)$ clearly satisfies the Dynamic HyperStructure axioms. Conversely, if $\varphi^{(m)} : \{*\} \times \mathcal{P}(H)^m \to \mathcal{P}(H) \times \{*\}$ is any Dynamic HyperStructure, then the maps

$$\star^{(m)}(A_1, \ldots, A_m) \ = \ \pi_1\big(\varphi^{(m)}(*, (A_1, \ldots, A_m))\big)$$

define an ordinary HyperStructure on $\mathcal{P}(H)$. □

Definition 10.11 (Dynamic n-SuperHyperStructure)**.** Let H, D, $\mathcal{I}$ be as above, and fix $n \geq 1$. Define iterated powersets by

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

A *Dynamic n-SuperHyperStructure* on (H, D) of signature $\mathcal{I}$ consists of, for each $m \in \mathcal{I}$, a map

$$\psi^{(m)} : D \times \big(\mathcal{P}^n(H)\big)^m \longrightarrow \mathcal{P}^n(H) \times D,$$

or its curried form

$$\psi^{c,(m)} : D \longrightarrow \underbrace{\mathcal{P}^n(H) \longrightarrow \cdots \longrightarrow \mathcal{P}^n(H)}_{m \text{ arrows}} \longrightarrow \mathcal{P}^n(H) \times D,$$

with the analogous compatibility condition.

Example 10.12 (Pallet Consolidation as a Dynamic 2-SuperHyperStructure)**.** Let H be the set of individual product identifiers (e.g. SKUs), and let

$$D = \text{List of log entries}$$

record consolidation actions. We view

$$\mathcal{P}^1(H) = \{\text{boxes of items}\}, \quad \mathcal{P}^2(H) = \{\text{pallets of boxes}\}.$$

Define the pallet-merging operation

$$\psi^{(2)}\big(d,\ (P_1, P_2)\big) = \Big(P_1\ \cup\ P_2,\ d\ \cup\ \{\text{“Merged pallets } P_1 \text{ and } P_2\text{”}\}\Big),$$

where $P_1, P_2 \subseteq \mathcal{P}^1(H)$ are two pallets. This returns a new pallet (the union of their box-sets) and appends a descriptive entry to the log. Thus $\{\psi^{(2)}\}$ forms a dynamic 2-SuperHyperStructure modeling hierarchical consolidation with stateful logging.

Theorem 10.13 (Dynamic n-SuperHyperStructure generalizes n-SuperHyperStructure)**.** *Every ordinary n-SuperHyperStructure* $\big(\mathcal{P}^n(H), \{\circ^{(m)} : (\mathcal{P}^n(H))^m \to \mathcal{P}^n(H)\}\big)$ *is realized by the trivial context* $D = \{*\}$ *and*

$$\psi^{(m)}(*, (X_1, \ldots, X_m)) = \big(\circ^{(m)}(X_1, \ldots, X_m), *\big).$$

Conversely, any Dynamic n-SuperHyperStructure on $(H, \{*\})$ *projects to an ordinary one.*

Proof. Identical to the previous proof, replacing $\mathcal{P}(H)$ by $\mathcal{P}^n(H)$ and $\star^{(m)}$ by $\circ^{(m)}$. □

Theorem 10.14 (Both Dynamic HyperStructure and Dynamic SuperHyperStructure generalize Dynamic Structure)**.** *Let* $\big(H, \{\theta^{(m)} : D \times H^m \to H \times D\}\big)$ *be any Dynamic Structure. Then:*

1. *It embeds into a Dynamic HyperStructure via*

$$\varphi^{(m)}\big(d, (\{x_1\}, \ldots, \{x_m\})\big) = \big(\{\ h\ \}, d'\big),$$

   where $(h, d') = \theta^{(m)}(d, (x_1, \ldots, x_m))$*, and extended arbitrarily on non-singleton inputs.*

2. *It embeds into a Dynamic n-SuperHyperStructure by identifying each* $x \in H$ *with the singleton chain* $\{\{\cdots\{x\}\cdots\}\} \in \mathcal{P}^n(H)$ *and defining* $\psi^{(m)}$ *so that*

$$\psi^{(m)}\big(d, (\iota(x_1), \ldots, \iota(x_m))\big) = \big(\iota(h), d'\big),$$

   where $\iota(x)$ *denotes the nested singleton and* $(h, d') = \theta^{(m)}(d, (x_1, \ldots, x_m))$.

Proof.

(1) Define $\varphi^{(m)} : D \times \mathcal{P}(H)^m \to \mathcal{P}(H) \times D$ by

$$\varphi^{(m)}(d, (A_1, \dots, A_m)) = \Big(\big\{\theta^{(m)}(d, (x_1, \dots, x_m))_1 \mid x_i \in A_i\big\},\ \theta^{(m)}(d, (x_1, \dots, x_m))_2\Big),$$

for any choice of $(x_1, \dots, x_m)$ in the A_i (all yield the same d' when θ is deterministic). On singleton inputs $A_i = \{x_i\}$, this reproduces $\theta^{(m)}$, so the original Dynamic Structure sits inside.

(2) Replace $\mathcal{P}(H)$ by $\mathcal{P}^n(H)$ and apply the same construction, noting that each $x \in H$ embeds as the n-fold singleton $\iota(x) \in \mathcal{P}^n(H)$.

Hence every Dynamic Structure appears as a special case of both Dynamic HyperStructure and Dynamic n-SuperHyperStructure. □

Funding

This study did not receive any financial or external support from organizations or individuals.

Acknowledgments

We extend our sincere gratitude to everyone who provided insights, inspiration, and assistance throughout this research. We particularly thank our readers for their interest and acknowledge the authors of the cited works for laying the foundation that made our study possible. We also appreciate the support from individuals and institutions that provided the resources and infrastructure needed to produce and share this paper. Finally, we are grateful to all those who supported us in various ways during this project.

Data Availability

This research is purely theoretical, involving no data collection or analysis. We encourage future researchers to pursue empirical investigations to further develop and validate the concepts introduced here.

Research Integrity

The authors affirm that, to the best of their knowledge, this manuscript represents their original research. It has not been previously published in any journal, nor is it currently being considered for publication elsewhere.

Disclaimer on Computational Tools

No computer-based tools—such as symbolic computation systems, automated theorem provers, or proof assistants (e.g., Mathematica, SageMath, Coq)—were employed in the development, analysis, or verification of the results contained in this paper. All derivations and proofs were conducted manually through analytical methods by the authors.

Ethical Approval

As this research is entirely theoretical in nature and does not involve human participants or animal subjects, no ethical approval is required.

Conflicts of Interest

The authors confirm that there are no conflicts of interest related to the research or its publication.

Code Availability

No code or software was developed for this study.

Clinical Trial

This study did not involve any clinical trials.

Consent to Participate

Not applicable.

Use of Generative AI and AI-Assisted Tools

We use generative AI and AI-assisted tools for tasks such as English grammar checking, and We do not employ them in any way that violates ethical standards.

Disclaimer

This work presents theoretical concepts that have not yet undergone practical testing or validation. Future researchers are encouraged to apply and assess these ideas in empirical contexts. While every effort has been made to ensure accuracy and appropriate referencing, unintentional errors or omissions may still exist. Readers are advised to verify referenced materials on their own. The views and conclusions expressed here are the authors' own and do not necessarily reflect those of their affiliated organizations.

References

[1] Madeleine Al-Tahan, Bijan Davvaz, Florentin Smarandache, and Osman Anis. On some neutrohyperstructures. *Symmetry*, 13(4):535, 2021.

[2] 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.

[3] 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.

[4] 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.

[5] Madeleine Al-Tahan and Bijan Davvaz. Chemical hyperstructures for elements with four oxidation states. *Iranian Journal of Mathematical Chemistry*, 13(2):85–97, 2022.

[6] Maria Santilli Ruggero and Thomas Vougiouklis. Hyperstructures in lie-santilli admissibility and iso-theories. *Ratio Mathematica*, 33:151, 2017.

[7] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[8] Alexander W. Kocurek. The logic of hyperlogic. part a: Foundations. *The Review of Symbolic Logic*, 17:244 – 271, 2022.

[9] Mark Burgin. Integrating random properties and the concept of probability. *Integration*, 3(2):137–155, 2012.

[10] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. In *Symposium on the Theory of Computing*, 1990.

[11] Jacob Burbea and C. Radhakrishna Rao. Entropy differential metric, distance and divergence measures in probability spaces: A unified approach. *Journal of Multivariate Analysis*, 12:575–596, 1982.

[12] Ferdinand Österreicher and Igor Vajda. A new class of metric divergences on probability spaces and its applicability in statistics. *Annals of the Institute of Statistical Mathematics*, 55:639–653, 2003.

[13] Jian Tang, Xinyang Feng, Bijan Davvaz, and Xiangyun Xie. A further study on ordered regular equivalence relations in ordered semihypergroups. *Open Mathematics*, 16:168 – 184, 2018.

[14] Muhammad Farooq, Asghar Khan, and Bijan Davvaz. Characterizations of ordered semihypergroups by the properties of their intersectional-soft generalized bi-hyperideals. *Soft Computing*, 22:3001–3010, 2018.

[15] Marc Krasner. A class of hyperrings and hyperfields. *International Journal of Mathematics and Mathematical Sciences*, 6:307–311, 1983.

[16] Fakhry Asad Agusfrianto, Saba Al-Kaseasbeh, Mariam Hariri, and Yudi Mahatma. On neutrohyperrings and neutroorderedhyperrings. *Neutrosophic Sets and Systems*, 77:1–19, 2025.

[17] 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.

[18] M Scafati Tallini. Hypervector spaces. In *Proceeding of the 4th International Congress in Algebraic Hyperstructures and Applications*, pages 167–174, 1991.

[19] Hans-Jurgen Bandelt and Victor Chepoi. Metric graph theory and geometry: a survey. *Contemporary Mathematics*, 453:49–86, 2008.

[20] Tomaz Pisanski and Milan Randic. Bridges between geometry and graph theory. In Catherine A. Gorini, editor, *Geometry at Work*, volume 53 of *MAA Notes*, pages 174–194. Cambridge University Press, 2000.

[21] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[22] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.

[23] FS Al-Anzi. Efficient cellular automata algorithms for planar graph and vlsi layout homotopic compaction. *International Journal of Computing and Information Sciences*, 1(1):1–17, 2003.

[24] John E Hopcroft and Jeffrey D Ullman. *Formal languages and their relation to automata*. Addison-Wesley Longman Publishing Co., Inc., 1969.

[25] Hüseyin Kamacı. Linguistic single-valued neutrosophic soft sets with applications in game theory. *International Journal of Intelligent Systems*, 36(8):3917–3960, 2021.

[26] Nicholas Kovach, Alan S. Gibson, and Gary B. Lamont. Hypergame theory: A model for conflict, misperception, and deception. 2015.

[27] Yutong Song and Yong Deng. Entropic explanation of power set. *International Journal of Computers, Communications & Control*, 16(4):4413, 2021.

[28] Yakar Kannai and Bezalel Peleg. A note on the extension of an order on a set to the power set. *Journal of Economic Theory*, 32(1):172–175, 1984.

[29] Souzana Vougioukli. Helix hyperoperation in teaching research. *Science & Philosophy*, 8(2):157–163, 2020.

[30] Souzana Vougioukli. Hyperoperations defined on sets of s-helix matrices. 2020.

[31] Akbar Rezaei, Florentin Smarandache, and S. Mirvakili. Applications of (neutro/anti)sophications to semihypergroups. *Journal of Mathematics*, 2021.

[32] Souzana Vougioukli. Helix-hyperoperations on lie-santilli admissibility. *Algebras Groups and Geometries*, 2023.

[33] Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

[34] Bijan Davvaz and Thomas Vougiouklis. *Walk Through Weak Hyperstructures, A: Hv-structures*. World Scientific, 2018.

[35] Piergiulio Corsini and Violeta Leoreanu. *Applications of hyperstructure theory*, volume 5. Springer Science & Business Media, 2013.

[36] Gulay Oguz and Bijan Davvaz. Soft topological hyperstructure. *J. Intell. Fuzzy Syst.*, 40:8755–8764, 2021.

[37] 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.

[38] F. Smarandache. Introduction to superhyperalgebra and neutrosophic superhyperalgebra. *Journal of Algebraic Hyperstructures and Logical Algebras*, 2022.

[39] Takaaki Fujita. Superhypermagma, lie superhypergroup, quotient superhypergroups, and reduced superhypergroups. *International Journal of Topology*, 2(3):10, 2025.

[40] Ajoy Kanti Das, Rajat Das, Suman Das, Bijoy Krishna Debnath, Carlos Granados, Bimal Shil, and Rakhal Das. A comprehensive study of neutrosophic superhyper bci-semigroups and their algebraic significance. *Transactions on Fuzzy Sets and Systems*, 8(2):80, 2025.

[41] Florentin Smarandache. Superhyperstructure & neutrosophic superhyperstructure, 2024. Accessed: 2024-12-01.

[42] Takaaki Fujita and Florentin Smarandache. Neutrosophic soft n-super-hypergraphs with real-world applications. *European Journal of Pure and Applied Mathematics*, 18(3):6621, 2025.

[43] Sunday Adesina Adebisi and Adetunji Patience Ajuebishi. The order involving the neutrosophic hyperstructures, the construction and setting up of a typical neutrosophic group. *HyperSoft Set Methods in Engineering*, 3:26–31, 2025.

[44] Takaaki Fujita. Entropy reimagined: Theoretical foundations of hyperentropy and superhyperentropy with real-world and engineering applications. *Spectrum of Engineering and Management Sciences*, 3(1):262–286, 2025.

[45] Abdullah Kargın and Memet Şahin. Superhyper groups and neutro–superhyper groups. *2023 Neutrosophic SuperHyperAlgebra And New Types of Topologies*, page 25, 2023.

[46] Florentin Smarandache. History of superhyperalgebra and neutrosophic superhyperalgebra (revisited again). *Neutrosophic Algebraic Structures and Their Applications*, page 10, 2022.

[47] Sirus Jahanpanah and Roohallah Daneshpayeh. An outspread on valued logic superhyperalgebras. *Facta Universitatis, Series: Mathematics and Informatics*, 2024.

[48] 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.

[49] Mohammad Hamidi, Florentin Smarandache, and Elham Davneshvar. Spectrum of superhypergraphs via flows. *Journal of Mathematics*, 2022(1):9158912, 2022.

[50] Takaaki Fujita. Hypergraph containers and their generalization to super-hyper-graph containers. *Abhath Journal of Basic and Applied Sciences*, 4(1):94–107, 2025.

[51] Takaaki Fujita. Chemical hypergraph and chemical superhypergraph. 2025.

[52] Eduardo Luciano Hernandez Ramos, Luis Ramiro Ayala Ayala, and Kevin Alexander Samaniego Macas. Study of factors that influence a victim's refusal to testify for sexual reasons due to external influence using plithogenic n-superhypergraphs. *Operational Research Journal*, 46(2):328–337, 2025.

[53] Min Huang, Fenghua Li, et al. Optimizing ai-driven digital resources in vocational english learning using plithogenic n-superhypergraph structures for adaptive content recommendation. *Neutrosophic Sets and Systems*, 88:283–295, 2025.

[54] 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.

[55] Florentin Smarandache. *Introduction to the n-SuperHyperGraph-the most general form of graph today*. Infinite Study, 2022.

[56] Adel Al-Odhari. A brief comparative study on hyperstructure, super hyperstructure, and n-super superhyperstructure. *Neutrosophic Knowledge*, 6:38–49, 2025.

[57] Adel Al-Odhari. Neutrosophic power-set and neutrosophic hyper-structure of neutrosophic set of three types. *Annals of Pure and Applied Mathematics*, 31(2):125–146, 2025.

[58] Takaaki Fujita, Maisam Jdid, and Florentin Smarandache. Hyperfunctions and superhyperfunctions in linear programming: Foundations and applications. *International Journal of Neutrosophic Science*, 26(4):65–76, 2025.

[59] Sirus Jahanpanah and Roohallah Daneshpayeh. On derived superhyper be-algebras. *Neutrosophic Sets and Systems*, 57(1):21, 2023.

[60] K Hila and T Vougiouklis. Weak hyperstructures in lie-santilli admissible. *Journal of Algebraic Hyperstructures and Logical Algebras*, pages 1–13, 2025.

[61] P Kamporoudi and T Vougiouklis. Cyclicity in some classes of hv-groups. *Journal of Algebraic Hyperstructures and Logical Algebras*, 1(3):91–97, 2020.

[62] 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.

[63] Shah Nawaz, Muhammad Gulistan, and Salma Khan. *Weak LA-hypergroups; neutrosophy, enumeration and redox reaction*, volume 36. Infinite Study, 2020.

[64] Takaaki Fujita. Chemical hyperstructures, superhyperstructures, and shv-structures: Toward a generalized framework for hierarchical chemical modeling. *ChemRxiv*, 2025.

[65] Takaaki Fujita. Hierarchical probability models: Tree and forest structures for treeprobability and forestprobability.

[66] Takaaki Fujita. Structural properties of algebraic treestructures and algebraic foreststructures.

[67] TK Shinoj and Sunil Jacob John. Intuitionistic fuzzy multisets and its application in medical diagnosis. *World academy of science, engineering and technology*, 6(1):1418–1421, 2012.

[68] PA Ejegwa. New operations on intuitionistic fuzzy multisets. *Journal of Mathematics and Informatics*, 3:17–23, 2015.

[69] KP Girish and Sunil Jacob John. Rough multiset and its multiset topology. In *Transactions on Rough Sets XIV*, pages 62–80. Springer, 2011.

[70] KP Girish and Sunil Jacob John. On rough multiset relations. *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, 3(4):306–326, 2014.

[71] Takaaki Fujita and Florentin Smarandache. A unified framework for structures and functorial structure: Managing super, hyper, superhyper, tree, and forest uncertain over/under/off models. *Neutrosophic Sets and Systems*, 91:337–380, 2025.

[72] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[73] 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.

[74] Hans-Jürgen Zimmermann. *Fuzzy set theory—and its applications*. Springer Science & Business Media, 2011.

[75] Humberto Bustince and P Burillo. Vague sets are intuitionistic fuzzy sets. *Fuzzy sets and systems*, 79(3):403–405, 1996.

[76] Krassimir Atanassov and George Gargov. Elements of intuitionistic fuzzy logic. part i. *Fuzzy sets and systems*, 95(1):39–52, 1998.

[77] Muhammad Akram, Bijan Davvaz, and Feng Feng. Intuitionistic fuzzy soft k-algebras. *Mathematics in Computer Science*, 7:353–365, 2013.

[78] Krassimir T Atanassov. Circular intuitionistic fuzzy sets. *Journal of Intelligent & Fuzzy Systems*, 39(5):5981–5986, 2020.

[79] K Tamilselvan, V Visalakshi, and Prasanalakshmi Balaji. Applications of picture fuzzy filters: performance evaluation of an employee using clustering algorithm. *AIMS Mathematics*, 8(9):21069–21088, 2023.

[80] Bui Cong Cuong and Vladik Kreinovich. Picture fuzzy sets-a new concept for computational intelligence problems. In *2013 third world congress on information and communication technologies (WICT 2013)*, pages 1–6. IEEE, 2013.

[81] R. Sivasamy and M. Mohammed Jabarulla. Products on interval valued pythagorean fuzzy soft graphs. *Journal of Interdisciplinary Mathematics*, 2024.

[82] Muhammad Sajjad Ali Khan, Saleem Abdullah, Asad Ali, Nasir Siddiqui, and Fazli Amin. Pythagorean hesitant fuzzy sets and their application to group decision making with incomplete weight information. *Journal of Intelligent & Fuzzy Systems*, 33(6):3971–3985, 2017.

[83] Yuping Xing, Runtong Zhang, Zhen Zhou, and Jun Wang. Some q-rung orthopair fuzzy point weighted aggregation operators for multi-attribute decision making. *Soft Computing*, 23:11627 – 11649, 2019.

[84] Peide Liu, Sumera Naz, Muhammad Akram, and Mamoona Muzammal. Group decision-making analysis based on linguistic q-rung orthopair fuzzy generalized point weighted aggregation operators. *International Journal of Machine Learning and Cybernetics*, 13:883 – 906, 2021.

[85] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Interval valued neutrosophic graphs. *Critical Review, XII*, 2016:5–33, 2016.

[86] S. Onar. A note on neutrosophic soft set over hyperalgebras. *Symmetry*, 16(10):1288, 2024.

[87] Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In *Philosophy*, pages 1–141. American Research Press, 1999.

[88] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Single valued neutrosophic graphs. *Journal of New theory*, (10):86–101, 2016.

[89] Said Broumi, Mohamed Talea, Assia Bakali, Florentin Smarandache, and PK Kishore Kumar. Shortest path problem on single valued neutrosophic graphs. In *2017 international symposium on networks, computers and communications (ISNCC)*, pages 1–6. IEEE, 2017.

[90] Florentin Smarandache, Memet Şahin, Derya Bakbak, Vakkas Uluçay, and Abdullah Kargın. *Neutrosophic SuperHyperAlgebra and New Types of Topologies*. Infinite Study, 2023.

[91] Shawkat Alkhazaleh. n-valued refined neutrosophic soft set theory. *Journal of Intelligent & Fuzzy Systems*, 32(6):4311–4318, 2017.

[92] Shawkat Alkhazaleh and Ayman A Hazaymeh. N-valued refined neutrosophic soft sets and their applications in decision making problems and medical diagnosis. *Journal of Artificial Intelligence and Soft Computing Research*, 8(1):79–86, 2018.

[93] Florentin Smarandache. n-valued refined neutrosophic logic and its applications to physics. *Infinite study*, 4:143–146, 2013.

[94] Florentin Smarandache. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*. Infinite study, 2018.

[95] Fazeelat Sultana, Muhammad Gulistan, Mumtaz Ali, Naveed Yaqoob, Muhammad Khan, Tabasam Rashid, and Tauseef Ahmed. A study of plithogenic graphs: applications in spreading coronavirus disease (covid-19) globally. *Journal of ambient intelligence and humanized computing*, 14(10):13139–13159, 2023.

[96] 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.

[97] Eduardo Martín Campoverde Valencia, Jessica Paola Chuisaca Vásquez, and Francisco Ángel Becerra Lois. Multineutrosophic analysis of the relationship between survival and business growth in the manufacturing sector of azuay province, 2020–2023, using plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 84:341–355, 2025.

[98] Julio Cesar Méndez Bravo, Claudia Jeaneth Bolanos Piedrahita, Manuel Alberto Méndez Bravo, and Luis Manuel Pilacuan-Bonete. Integrating smed and industry 4.0 to optimize processes with plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 84:328–340, 2025.

[99] John N Mordeson and Premchand S Nair. *Fuzzy graphs and fuzzy hypergraphs*, volume 46. Physica, 2012.

[100] M Vijaya and M Asha Joyce. Operations on interval-valued binary fuzzy graphs. *Journal of Algebraic Statistics*, 13(2):2534–2540, 2022.

[101] Dalibor Krleža and Krešimir Fertalj. Graph matching using hierarchical fuzzy graph neural networks. *Ieee transactions on fuzzy systems*, 25(4):892–904, 2016.

[102] Azriel Rosenfeld. Fuzzy graphs. In *Fuzzy sets and their applications to cognitive and decision processes*, pages 77–95. Elsevier, 1975.

[103] Ahmed Hatip et al. On the two-fold neutrosophic groups and their algebra properties. *Journal of Neutrosophic and Fuzzy Systems*, (1):28–8, 2024.

[104] John Baez and Mike Stay. Physics, topology, logic and computation: a rosetta stone. In *New structures for physics*, pages 95–172. Springer, 2010.

[105] John C Reynolds. Definitional interpreters for higher-order programming languages. In *Proceedings of the ACM annual conference-Volume 2*, pages 717–740, 1972.

[106] Henk P Barendregt. Introduction to lambda calculus. 1984.

[107] Graham Hutton. *Programming in haskell*. Cambridge University Press, 2016.

[108] Paul Keir, Andrew Gozillon, and Seyed Hossein Haeri. Intrinsic currying for c++ template metaprograms. In *International Symposium on Trends in Functional Programming*, pages 46–73. Springer, 2018.

[109] Paul Keir[1], Andrew Gozillon[1], and Seyed Hossein Haeri. Intrinsic currying for c++ template. In *Trends in Functional Programming: 19th International Symposium, TFP 2018, Gothenburg, Sweden, June 11–13, 2018, Revised Selected Papers*, volume 11457, page 46. Springer, 2019.

[110] Wolfgang Lux. Adding haskell-style overloading to curry. In *Workshop of Working Group*, volume 2, pages 67–76, 2008.

[111] Antony JT Davie. *Introduction to Functional Programming Systems Using Haskell*. Number 27. Cambridge University Press, 1992.

[112] Takaaki Fujita. How to represent a$\rightarrow$ b $\rightarrow \cdots \rightarrow$ z: From curried functions and hyperfunctions to curried structures and hyperstructures, and more, 2025.

[113] Grzegorz Bancerek. Curried and uncurried functions. *Formalized Mathematics*, 1(3):537–541, 1990.