Property HyperGraphs and Property SuperHyperGraphs for Data Analysis

Takaaki Fujita [1] *

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

Abstract

Graph theory provides a rigorous mathematical foundation for modeling relationships by representing entities as vertices and their interactions as edges [1,2]. Hypergraphs generalize this paradigm by allowing hyperedges to connect arbitrary subsets of vertices [3], and SuperHyperGraphs extend it further via iterated powerset constructions that capture hierarchical, multi-layer linkages among edges [4, 5]. These enriched models support applications across biology, social networks, signal processing, and knowledge representation.

Property Graphs are directed multigraphs in which vertices and edges carry key–value properties and edges additionally bear labels, enabling schema-flexible modeling of heterogeneous, real-world data(cf. [6–8]). In this paper, we show how to elevate Property Graphs to the settings of HyperGraphs and SuperHyperGraphs by introducing formal definitions for Property HyperGraphs and Property SuperHyperGraphs and presenting preliminary theoretical results that demonstrate their expressive power.

Keywords: Property Graphs, Superhypergraphs, Hypergraphs, Property SuperHyperGraphs, Property HyperGraphs

1 Preliminaries

Throughout this paper, we adopt a consistent vocabulary and notation. Unless otherwise noted, all graphs considered are finite. For further background on less familiar operations or concepts, the interested reader is referred to the cited literature.

1.1 SuperHyperGraphs

A finite *hypergraph* generalizes the classical graph model by permitting *hyperedges* that connect any non-empty subset of vertices [9–11, 11–13]. Building on this concept, a finite *SuperHyperGraph* is obtained by iteratively applying the powerset operator, thereby creating nested hierarchies of vertex and edge sets that encode multi-layered relationships [5, 14–20]. Such structures have demonstrated utility in areas ranging from molecular design and complex-network analysis to advanced signal-processing pipelines [18, 21, 22]. Unless stated otherwise, the integer n in $P_n(\cdot)$ or in an n-SuperHyperGraph is assumed to be non-negative.

Definition 1.1 (Base Set)**.** A *base set* S is the initial universe of discourse:

$$S = \{ x \mid x \text{ belongs to the context at hand} \}.$$

Every element that appears in $\mathcal{P}(S)$ or in any iterated powerset $\mathcal{P}_n(S)$ must of course lie in S.

Definition 1.2 (Powerset)**.** For a set S, the *powerset* $\mathcal{P}(S)$ is the family of all subsets of S:

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

This collection includes both S itself and the empty set $\emptyset$.

Definition 1.3 (Hypergraph)**.** [3, 23] A *hypergraph* is an ordered pair $H = (V, E)$ where

- V is a finite vertex set, and
- E is a finite family of non-empty subsets of V; the members of E are called *hyperedges*.

Hypergraphs naturally represent interactions that involve more than two participants.

Example 1.4 (Hypergraph Model for Market-Basket Analysis)**.** Market-basket analysis seeks to uncover groups of products that are purchased together. A compact way to represent the entire transaction log is to view it as a *hypergraph* in the sense of Definition 1.3.

Vertex set. Let

$$V = \{\,Bread, Butter, Milk, Eggs, Coffee, Cheese\,\}$$

be the collection of six distinct items sold in a small grocery store.

Observed transactions (hyperedges). During one afternoon the cash register records four baskets:

$$\begin{aligned} T_1 &= \{Bread, Butter, Milk\}, \\ T_2 &= \{Coffee, Milk\}, \\ T_3 &= \{Bread, Eggs, Cheese\}, \\ T_4 &= \{Butter, Eggs, Milk\}. \end{aligned}$$

Each transaction is a non-empty subset of V and is therefore admissible as a hyperedge.

Resulting hypergraph. Define

$$E = \{\,T_1, T_2, T_3, T_4\,\}.$$

The pair $H = (V, E)$ is a hypergraph whose hyperedges correspond one-to-one with the observed baskets.

Mining tasks enabled by this representation

- *Support counting:* the degree of a vertex equals the number of baskets that contain the corresponding item, providing its purchase frequency.
- *Frequent-itemset discovery:* any subset of vertices that appears together in a sufficient number of hyperedges constitutes a frequent pattern.
- *Hyperedge clustering:* grouping baskets that share many items can reveal customer segments with similar buying habits.
- *Edge contraction for category analysis:* by merging vertices into higher-level product categories (e.g. dairy, bakery), the same hypergraph can be coarsened without revisiting the raw data.

The hypergraph model thus captures all transactions simultaneously while retaining enough structure to support the full spectrum of basket-mining algorithms, from simple frequency counts to sophisticated community detection among both items and customers.

Definition 1.5 (n-th Powerset)**.** [24–28] Let X be a set. The first powerset is $\mathcal{P}_1(X) = \mathcal{P}(X)$. For $n \geq 1$ we define

$$\mathcal{P}_{n+1}(X) = \mathcal{P}\big(P_n(X)\big).$$

When the empty set is excluded one writes $\mathcal{P}_n^*(X) = \mathcal{P}_n(X) \setminus \{\emptyset\}$.

Example 1.6 (n-th Powerset in Market-Basket Data Mining)**.** Suppose a supermarket tracks customer purchases over a single week and observes the following five items:

$$X = \{Bread, Milk, Eggs, Butter, Coffee\}.$$

$P_1(X) = \mathcal{P}(X)$: candidate itemsets. Every non-empty subset of X is a *candidate itemset*. For instance, $\{\,Bread, Butter\}$ or $\{\,Milk, Eggs, Coffee\}$. Standard algorithms such as Apriori scan the transaction log to determine which of these subsets occur frequently.

$P_2(X) = \mathcal{P}\big(P_1(X)\big)$: clusters of itemsets. The second powerset groups itemsets into *itemset clusters*. An analyst may, for example, collect all frequent two-item combinations that involve *Coffee*:

$$C_{Coffee} = \big\{\{Coffee, Milk\}, \{Coffee, Bread\}, \{Coffee, Eggs\}\big\} \in P_2(X).$$

$P_3(X) = \mathcal{P}(P_2(X))$: meta-clusters. The third powerset organises clusters of itemsets into *meta-clusters*. One might place all itemset clusters whose underlying products form a typical *breakfast* assortment into a single meta-cluster:

$$M_{Breakfast} = \{C_{Coffee}, C_{Milk}, C_{Eggs}\} \in P_3(X).$$

Interpretation.

- Level 1 ($P_1(X)$) supports traditional frequent-itemset mining.
- Level 2 ($P_2(X)$) enables discovery of *correlated patterns*, such as sets of itemsets that often appear together across many market segments.
- Level 3 ($P_3(X)$) facilitates higher-order reasoning, e.g. comparing entire pattern families between different seasons or geographical regions.

In practical pipelines, each higher powerset serves as the search space for progressively more abstract data-mining tasks: association-rule discovery at level 1, pattern clustering at level 2, and meta-pattern comparison or visual analytics at level 3 and beyond.

Definition 1.7 (n-SuperHyperGraph)**.** [29–31] Let V_0 be a finite base set. Define iteratively

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

An *n-SuperHyperGraph* is a pair

$$\mathrm{SHG}^{(n)} = (V, E), \quad V, E \subseteq \mathcal{P}^n(V_0),$$

where each element of V is called an n-supervertex and each element of E an n-superedge.

Example 1.8 (A 2-SuperHyperGraph for Market-Basket Data Mining)**.** A classical task in data mining is the discovery of *frequent itemsets* from a collection of customer transactions. We show how such a dataset can be organised as a 2-SuperHyperGraph in the sense of Definition 1.7.

Step 0 — Base set of items. Assume the shop sells four items $V_0 = \{A, B, C, D\}$, where A = bread, B = milk, C = eggs, D = butter.

Step 1 — First powerset: transactions. The first iterated powerset $\mathcal{P}^1(V_0) = \mathcal{P}(V_0)$ contains all possible baskets. Select four actual transactions and regard them as *1-supervertices*:

$$T_1 = \{A, B\}, \quad T_2 = \{B, C, D\}, \quad T_3 = \{A, C, D\}, \quad T_4 = \{A, B, D\} \in \mathcal{P}^1(V_0).$$

Step 2 — Second powerset: pattern groups. The second powerset $\mathcal{P}^2(V_0) = \mathcal{P}(\mathcal{P}(V_0))$ collects *sets of transactions*. Define three *2-supervertices* that represent candidate frequent item-set groups:

$$V = \Big\{G_1 = \{T_1, T_4\},\ G_2 = \{T_2, T_3\},\ G_3 = \{T_2, T_3, T_4\}\Big\} \subseteq \mathcal{P}^2(V_0).$$

- G_1 groups baskets that both contain $\{A, B\}$.
- G_2 gathers baskets containing $\{C, D\}$.
- G_3 captures baskets containing $\{D\}$ together with at least one additional item.

Step 3 — 2-superedges. Define the 2-superedge family

$$E = \Big\{E_1 = \{G_1, G_3\},\ E_2 = \{G_2, G_3\}\Big\} \subseteq \mathcal{P}(V) \setminus \{\varnothing\}.$$

- E_1 links pattern groups whose underlying baskets share the common sub-itemset $\{A, B, D\}$.
- E_2 links pattern groups sharing the sub-itemset $\{C, D\}$.

Resulting 2-SuperHyperGraph.

$$\mathrm{SHG}^{(2)} \;=\; (V, E) \quad \text{with } V, E \subseteq \mathcal{P}^2(V_0).$$

Vertices at level 2 capture candidate frequent patterns; edges capture higher-order correlations among those patterns. Analytic algorithms (e.g. level-wise Apriori) may now operate directly on $\mathrm{SHG}^{(2)}$ to prune infrequent groups or to generate association rules such as $\{A, B\} \Rightarrow D$ whenever E_1 is deemed sufficiently strong.

The construction naturally generalises: further powerset iterations ($n > 2$) would allow nested clusters of pattern groups, supporting multi-resolution pattern mining in large-scale transactional corpora.

Example 1.9 (A 3-SuperHyperGraph for Multi-Tier Database Development)**.** Modern cloud applications often organise data in multiple architectural tiers—tables, schemas (logical databases), and service-level groupings. Definition 1.7 naturally captures this hierarchy.

Step 0 — Base set (V_0): atomic columns.

$$V_0 = \{\mathit{UserID}, \mathit{Name}, \mathit{Email}, \mathit{OrderID}, \mathit{OrderDate}, \mathit{ProductID}, \mathit{Quantity}, \mathit{Price}\}.$$

Step 1 — $\mathcal{P}^1(V_0)$: relational tables (1-supervertices).

$$\begin{aligned} T_{\mathsf{Users}} &= \{\mathit{UserID}, \mathit{Name}, \mathit{Email}\}, \\ T_{\mathsf{Orders}} &= \{\mathit{OrderID}, \mathit{UserID}, \mathit{OrderDate}\}, \\ T_{\mathsf{OrderItems}} &= \{\mathit{OrderID}, \mathit{ProductID}, \mathit{Quantity}, \mathit{Price}\}. \end{aligned}$$

We take $V_1 = \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\} \subseteq \mathcal{P}^1(V_0)$.

1-superedges.

$$E_1 = \big\{\{T_{\mathsf{Users}}, T_{\mathsf{Orders}}\},\ \{T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\}\big\} \subseteq \mathcal{P}(V_1) \setminus \{\varnothing\},$$

encoding the two foreign-key relationships (*Orders.UserID* $\to$ *Users.UserID* and *OrderItems.OrderID* $\to$ *Orders.OrderID*).

Step 2 — $\mathcal{P}^2(V_0)$: logical databases (2-supervertices).

$$\begin{aligned} \mathsf{EComDB} &= \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\}, \\ \mathsf{AnalyticsDB} &= \{T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\}. \end{aligned}$$

Thus $V_2 = \{\mathsf{EComDB}, \mathsf{AnalyticsDB}\} \subseteq \mathcal{P}^2(V_0)$.

2-superedge. $E_2 = \big\{\{\mathsf{EComDB}, \mathsf{AnalyticsDB}\}\big\}$, reflecting an ETLjob that copies data nightly from the operational database into an analytical replica.

Step 3 — $\mathcal{P}^3(V_0)$: service clusters (3-supervertices).

$$\begin{aligned} \mathsf{TransactionSvc} &= \{\mathsf{EComDB}\}, \\ \mathsf{ReportingSvc} &= \{\mathsf{AnalyticsDB}\}. \end{aligned}$$

Hence $V_3 = \{\mathsf{TransactionSvc}, \mathsf{ReportingSvc}\} \subseteq \mathcal{P}^3(V_0)$.

3-superedge.

$$E_3 = \big\{\{\mathsf{TransactionSvc}, \mathsf{ReportingSvc}\}\big\},$$

representing an *event stream* (e.g. Kafka topic) through which the transaction service publishes updates that the reporting service consumes.

Resulting 3-SuperHyperGraph.

$$\mathrm{SHG}^{(3)} = (V_3, E_3),$$
$$\text{with } V_3, E_3 \subseteq \mathcal{P}^3(V_0).$$

Interpretation.

- Level 0 (columns) captures atomic data fields.
- Level 1 (tables) models relational schemas and primary/foreign-key constraints.
- Level 2 (databases) expresses logical groupings deployed as distinct schemas or instances.
- Level 3 (services) groups databases into microservices, together with inter-service communication edges.

Operations such as schema evolution, service refactoring, or dependency analysis can therefore be formalised as transformations or traversals on $\mathrm{SHG}^{(3)}$, enabling rigorous reasoning about complex multi-tier database architectures.

1.2 Property Graph

A Property Graph is a directed multigraph whose vertices and edges carry arbitrary key–value attributes, and edges additionally possess labels, enabling schema-flexible relational data modelling [6–8, 32–38].

Definition 1.10 (Property Graph)**.** Fix three (possibly infinite) sets

$$\Sigma \quad \text{(edge–label alphabet)},$$
$$K \quad \text{(property keys)},$$
$$S \quad \text{(property values)}.$$

A *property graph*[1] is a septuple

$$G = \big(V,\ E,\ s,\ t,\ \lambda,\ \mu,\ \bot\big)$$

whose components satisfy the following conditions:

(a) V is a finite (or at most countable) set whose elements are called *vertices* (or *nodes*).

(b) E is a finite (or at most countable) set whose elements are called *edges*. Distinct edges may share the same endpoints, so (V, E) is a *multigraph*.

(c) $s, t : E \to V$ are the *source* and *target* functions. For $e \in E$ we write $s(e) \xrightarrow{e} t(e)$.

(d) $\lambda : E \to \Sigma$ assigns a label (drawn from the alphabet Σ) to every edge.

(e) $\mu : (V \cup E) \times K \longrightarrow S \cup \{\bot\}$ is the *property map*. For an entity $x \in V \cup E$ and a key $k \in K$, the value $\mu(x, k)$ is either a member of S or the distinguished symbol $\bot$ indicating that x has no value for key k.

(f) The symbol $\bot \notin S$ is fixed once and for all and is **not** considered a valid property value.

We write

$$\mathrm{keyset}(x) := \{\, k \in K \mid \mu(x, k) \neq \bot\}, \qquad \mathrm{val}(x, k) := \mu(x, k) \quad (k \in \mathrm{keyset}(x)),$$

and call $\langle k, \mu(x, k)\rangle$ an *attribute* of x.

Remark 1.11. (i) Allowing E to be a multiset (or, equivalently, introducing edge identifiers) lets two vertices be joined by arbitrarily many edges—even with identical labels and attributes.

(ii) If λ is constant (all edges share one label) and $\mu \equiv \bot$, Definition 1.10 collapses to the usual concept of a directed multigraph.

(iii) Many graph-database operations (e.g. Gremlin traversals) can be formalised as functions $T : \mathcal{P}(V) \to \mathcal{P}(V)$; see Yamaguchi *et al.* for a standard treatment.

Example 1.12 (Property Graph for a Streaming-Platform Dataset)**.** We illustrate Definition 1.10 with a concrete, small-scale model of an on-line movie-streaming service.

[1] Equivalent to "directed, edge-labelled, attributed multigraph" in the graph-database literature.

Label alphabet, keys, and value domain.

$$\Sigma = \{\texttt{FOLLOWS}, \texttt{RATED}, \texttt{HAS\_GENRE}\},$$

$$K = \{\texttt{type}, \texttt{name}, \texttt{age}, \texttt{city}, \texttt{title}, \texttt{year}, \texttt{rating}, \texttt{date}\},$$

$$S = \mathbb{N} \cup \mathbb{R} \cup \text{Strings}.$$

Vertices. The vertex set

$$V = \{p_1, p_2, m_1, m_2, g_1\}$$

is partitioned by the attribute `type`:

$$\mu(p_i, \texttt{type}) = \text{``Person''} \ (i = 1, 2),$$

$$\mu(m_j, \texttt{type}) = \text{``Movie''} \ (j = 1, 2),$$

$$\mu(g_1, \texttt{type}) = \text{``Genre''}.$$

Selected vertex attributes are

$$\begin{array}{rclrcl} \mu(p_1, \texttt{name}) & = & \text{``Alice''} & \mu(p_2, \texttt{name}) & = & \text{``Bob''} \\ \mu(p_1, \texttt{age}) & = & 27 & \mu(p_2, \texttt{age}) & = & 25 \\ \mu(p_1, \texttt{city}) & = & \text{``Tokyo''} & \mu(p_2, \texttt{city}) & = & \text{``Kyoto''} \\ \mu(m_1, \texttt{title}) & = & \text{``Inception''} & \mu(m_1, \texttt{year}) & = & 2010 \\ \mu(m_2, \texttt{title}) & = & \text{``Interstellar''} & \mu(m_2, \texttt{year}) & = & 2014 \\ \mu(g_1, \texttt{name}) & = & \text{``Sci-Fi''}. & & & \end{array}$$

Edges, endpoints, labels.

$$\begin{array}{rcllr} E & = & \{e_1, e_2, e_3, e_4, e_5\}, & & \\ s(e_1) & = & p_1, & t(e_1) = p_2, & \lambda(e_1) = \texttt{FOLLOWS}, \\ s(e_2) & = & p_1, & t(e_2) = m_1, & \lambda(e_2) = \texttt{RATED}, \\ s(e_3) & = & p_1, & t(e_3) = m_2, & \lambda(e_3) = \texttt{RATED}, \\ s(e_4) & = & m_1, & t(e_4) = g_1, & \lambda(e_4) = \texttt{HAS\_GENRE}, \\ s(e_5) & = & m_2, & t(e_5) = g_1, & \lambda(e_5) = \texttt{HAS\_GENRE}. \end{array}$$

Edge attributes.

$$\mu(e_1, \texttt{date}) = \text{``2025-05-12''}, \quad \mu(e_2, \texttt{rating}) = 5,$$

$$\mu(e_2, \texttt{date}) = \text{``2025-05-13''}, \ \mu(e_3, \texttt{rating}) = 4,$$

$$\mu(e_3, \texttt{date}) = \text{``2025-05-14''}.$$

All other $\mu(x, k)$ not listed are set to the distinguished value $\perp$.

The septuple $G = (V, E, s, t, \lambda, \mu, \perp)$ thus obtained satisfies every clause of Definition 1.10. It captures users (*persons*), movies, and genres as vertices; user-to-user `FOLLOWS` relationships, user ratings of movies, and movie-to-genre links as labelled, attributed edges. Additional properties or vertex types (e.g. `Director`, `Studio`) can be incorporated seamlessly by extending the key set K and adding new vertices and edges.

2 Results and Revisits: Property HyperGraphs

Property HyperGraphs have been examined in several earlier studies; in this paper we revisit and analyse them in greater depth.

Fix three (possibly infinite) sets

$$\Sigma \quad \text{(hyperedge-label alphabet)}, \qquad K \quad \text{(property keys)}, \qquad S \quad \text{(property values)},$$

and let $\bot \notin S$ be a distinguished symbol.

Definition 2.1 (Property HyperGraph)**.** A *property hypergraph* is a quadruple

$$H = \big(V,\, E,\, \lambda,\, \mu\big)$$

satisfying:

(a) V is a finite (or at most countable) set of *vertices*.

(b) E is a finite family of non-empty subsets of V, called *hyperedges*.

(c) $\lambda : E \to \Sigma$ assigns to each hyperedge a label.

(d) $\mu : (V \cup E) \times K \to S \cup \{\bot\}$ is the *property map*, where

$$\mu(x, k) = \begin{cases} s \in S, & \text{if } x \text{ has property } k \text{ with value } s, \\ \bot, & \text{if no value is assigned.} \end{cases}$$

We write

$$\operatorname{keyset}(x) := \{\, k \in K \mid \mu(x, k) \neq \bot \},\qquad \operatorname{val}(x, k) := \mu(x, k) \quad (k \in \operatorname{keyset}(x)).$$

Example 2.2 (Property HyperGraph for Patient–Symptom Dataset)**.** We illustrate Definition 2.1 by modelling a clinical dataset in which each patient record links the symptoms they exhibit and carries patient metadata.

Label alphabet, keys, and value domain.

$$\Sigma = \{\texttt{COVID19}, \texttt{Influenza}, \texttt{Migraine}\},$$

$$K = \{\texttt{category}, \texttt{ICD\_code}, \texttt{age}, \texttt{gender}, \texttt{severity}\},$$

$$S = \{\text{``Constitutional''}, \text{``Respiratory''}, \text{``Neurological''}\} \cup \{\text{Strings}\} \cup \mathbb{N}.$$

Vertices (symptoms). Let

$$V = \{v_1, v_2, v_3, v_4, v_5\},$$

where

$$\mu(v_1, \texttt{category}) = \text{``Constitutional''}, \quad \mu(v_1, \texttt{ICD\_code}) = \text{``R50.9''},$$
$$\mu(v_2, \texttt{category}) = \text{``Respiratory''}, \quad \mu(v_2, \texttt{ICD\_code}) = \text{``R05''},$$
$$\mu(v_3, \texttt{category}) = \text{``Constitutional''}, \quad \mu(v_3, \texttt{ICD\_code}) = \text{``R53.83''},$$
$$\mu(v_4, \texttt{category}) = \text{``Neurological''}, \quad \mu(v_4, \texttt{ICD\_code}) = \text{``R51''},$$
$$\mu(v_5, \texttt{category}) = \text{``Respiratory''}, \quad \mu(v_5, \texttt{ICD\_code}) = \text{``R06.02''}.$$

Hyperedges (patient records). Define three patient hyperedges:

$$E = \{e_1, e_2, e_3\},$$

with

$$e_1 = \{v_1, v_2, v_3\}, \quad e_2 = \{v_2, v_3, v_4\}, \quad e_3 = \{v_1, v_4, v_5\}.$$

Labels and properties. Assign each record a diagnosis label and patient metadata:

$$\lambda(e_1) = \texttt{COVID19}, \quad \mu(e_1, \texttt{age}) = 45, \quad \mu(e_1, \texttt{gender}) = \text{``Male''}, \quad \mu(e_1, \texttt{severity}) = \text{``Moderate''},$$

$$\lambda(e_2) = \texttt{Influenza}, \quad \mu(e_2, \texttt{age}) = 30, \quad \mu(e_2, \texttt{gender}) = \text{``Female''}, \quad \mu(e_2, \texttt{severity}) = \text{``Mild''},$$

$$\lambda(e_3) = \texttt{Migraine}, \quad \mu(e_3, \texttt{age}) = 25, \quad \mu(e_3, \texttt{gender}) = \text{``Female''}, \quad \mu(e_3, \texttt{severity}) = \text{``Severe''}.$$

Keysets and values. For example,

$$\text{keyset}(v_4) = \{\texttt{category}, \texttt{ICD\_code}\}, \quad \text{val}(v_4, \texttt{ICD\_code}) = \text{``R51''},$$

$$\text{keyset}(e_2) = \{\texttt{age}, \texttt{gender}, \texttt{severity}\}, \quad \text{val}(e_2, \texttt{severity}) = \text{``Mild''}.$$

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies Definition 2.1, modelling a patient-symptom dataset in which each record links multiple symptoms and carries patient attributes.

Example 2.3 (Property HyperGraph for University Course Enrollment)**.** We illustrate Definition 2.1 by modelling a university's course-enrollment system.

Label alphabet, keys, and value domain.

$$\Sigma = \{\texttt{CSE101}, \texttt{MATH202}, \texttt{HIST303}\},$$

$$K = \{\texttt{role}, \texttt{name}, \texttt{semester}, \texttt{credits}\},$$

$$S = \{\text{``Instructor''}, \text{``Student''}, \text{``TA''}\} \cup \{\text{Strings}\} \cup \mathbb{N}.$$

Vertices. Let

$$V = \{ v_1, v_2, v_3, v_4, v_5 \}$$

where

$$\mu(v_1, \texttt{name}) = \text{``Dr. Smith''}, \quad \mu(v_1, \texttt{role}) = \text{``Instructor''},$$

$$\mu(v_2, \texttt{name}) = \text{``Alice''}, \quad \mu(v_2, \texttt{role}) = \text{``Student''},$$

$$\mu(v_3, \texttt{name}) = \text{``Bob''}, \quad \mu(v_3, \texttt{role}) = \text{``Student''},$$

$$\mu(v_4, \texttt{name}) = \text{``Carol''}, \quad \mu(v_4, \texttt{role}) = \text{``TA''},$$

$$\mu(v_5, \texttt{name}) = \text{``Dave''}, \quad \mu(v_5, \texttt{role}) = \text{``Student''}.$$

Hyperedges. Define three hyperedges, one per course:

$$E = \left\{ e_1, e_2, e_3 \right\},$$

with

$$e_1 = \{v_1, v_2, v_3, v_4\}, \quad e_2 = \{v_1, v_2, v_5\}, \quad e_3 = \{v_1, v_3, v_5\}.$$

Labels and properties. For each course hyperedge e_i we set

$$\lambda(e_1) = \texttt{CSE101}, \quad \lambda(e_2) = \texttt{MATH202}, \quad \lambda(e_3) = \texttt{HIST303},$$

and assign

$$\mu(e_1, \texttt{semester}) = \text{``Fall 2025''}, \quad \mu(e_1, \texttt{credits}) = 4,$$

$$\mu(e_2, \texttt{semester}) = \text{``Spring 2025''}, \quad \mu(e_2, \texttt{credits}) = 3,$$

$$\mu(e_3, \texttt{semester}) = \text{``Fall 2025''}, \quad \mu(e_3, \texttt{credits}) = 3.$$

Keysets and values. For example,

$$\text{keyset}(v_2) = \{\texttt{name}, \texttt{role}\}, \quad \text{val}(v_2, \texttt{name}) = \text{``Alice''},$$

$$\text{keyset}(e_1) = \{\texttt{semester}, \texttt{credits}\}, \quad \text{val}(e_1, \texttt{credits}) = 4.$$

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies all clauses of Definition 2.1: Vertices represent people with roles, hyperedges represent courses linking instructor, students, and TAs, each course carries a label (course code) and properties (semester, credits).

Example 2.4 (Property HyperGraph for Film Productions)**.** We illustrate Definition 2.1 by modelling a film-production scenario.

Label alphabet, keys, and value domain.

$$\Sigma = \{\texttt{TheGreatAdventure}, \texttt{MysteryNight}\},$$

$$K = \{\texttt{roleType}, \texttt{name}, \texttt{birthYear}, \texttt{nationality}, \texttt{releaseYear}, \texttt{genre}, \texttt{boxOffice}\},$$

$$S = \{\text{``Actor''}, \text{``Director''}, \text{``Producer''}\} \cup \text{Strings} \cup \mathbb{N}.$$

Vertices. Let

$$V = \{ v_1, v_2, v_3, v_4, v_5\},$$

with properties

$$\mu(v_1, \texttt{name}) = \text{``Alice Johnson''}, \quad \mu(v_1, \texttt{roleType}) = \text{``Actor''}, \quad \mu(v_1, \texttt{birthYear}) = 1985,$$

$$\mu(v_2, \texttt{name}) = \text{``Bob Lee''}, \quad \mu(v_2, \texttt{roleType}) = \text{``Actor''}, \quad \mu(v_2, \texttt{birthYear}) = 1978,$$

$$\mu(v_3, \texttt{name}) = \text{``Carol Smith''}, \quad \mu(v_3, \texttt{roleType}) = \text{``Director''}, \quad \mu(v_3, \texttt{nationality}) = \text{``USA''},$$

$$\mu(v_4, \texttt{name}) = \text{``David Kumar''}, \quad \mu(v_4, \texttt{roleType}) = \text{``Producer''}, \quad \mu(v_4, \texttt{nationality}) = \text{``UK''},$$

$$\mu(v_5, \texttt{name}) = \text{``Eva Zhang''}, \quad \mu(v_5, \texttt{roleType}) = \text{``Actor''}, \quad \mu(v_5, \texttt{birthYear}) = 1990.$$

Hyperedges. Define two film hyperedges:

$$E = \{ e_1, e_2\},$$

where

$$e_1 = \{v_1, v_2, v_3, v_4\} \quad \text{(The Great Adventure cast/crew)},$$

$$e_2 = \{v_2, v_3, v_5\} \quad \text{(Mystery Night cast/crew)}.$$

Labels and properties. For each film hyperedge:

$$\lambda(e_1) = \texttt{TheGreatAdventure}, \quad \mu(e_1, \texttt{releaseYear}) = 2024,$$

$$\mu(e_1, \texttt{genre}) = \text{``Action''}, \quad \mu(e_1, \texttt{boxOffice}) = 120000000,$$

$$\lambda(e_2) = \texttt{MysteryNight}, \quad \mu(e_2, \texttt{releaseYear}) = 2023,$$

$$\mu(e_2, \texttt{genre}) = \text{``Mystery''}, \quad \mu(e_2, \texttt{boxOffice}) = 85000000.$$

Keysets and values. For example,

$$\text{keyset}(v_3) = \{\texttt{name}, \texttt{roleType}, \texttt{nationality}\},$$

$$\text{val}(v_3, \texttt{roleType}) = \text{``Director''},$$

$$\text{keyset}(e_1) = \{\texttt{releaseYear}, \texttt{genre}, \texttt{boxOffice}\},$$

$$\text{val}(e_1, \texttt{genre}) = \text{``Action''}.$$

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies all requirements of Definition 2.1: vertices represent cast and crew with personal attributes; hyperedges represent films linking multiple participants, each carrying a label (film title) and properties (release year, genre, box office).

Theorem 2.5 (Generalisation of Property Graphs and Hypergraphs)**.** *Let $H = (V, E, \lambda, \mu)$ be a property hypergraph over $(\Sigma, K, S, \perp)$. Then:*

(i) If every hyperedge $e \in E$ satisfies $|e| = 2$ and we equip $e = \{u, v\}$ with an arbitrary orientation $u \to v$, then $(V, E, s, t, \lambda, \mu, \perp)$ is precisely a Property Graph as in Definition 1.10.

(ii) If $\Sigma = \{\sigma_0\}$ is a singleton and $\mu(x, k) \equiv \perp$ for all (x, k), then $H = (V, E)$ collapses to an ordinary Hypergraph as in Definition 1.3.

Proof. For (i), restrict every 2-element hyperedge $e = \{u, v\}$ to a directed edge by choosing one of the two orderings (u, v) or (v, u). The label map $\lambda : E \to \Sigma$ and the property map μ coincide with those of a Property Graph. All axioms (a)–(f) of Definition 1.10 follow immediately.

For (ii), since Σ has only one element, λ carries no additional information; and because $\mu \equiv \perp$, no vertex or hyperedge carries a property. Thus $H = (V, E)$ satisfies exactly the conditions of Definition 1.3, concluding the proof. □

Remark 2.6. Theorem 2.5 shows that Property HyperGraphs strictly extend both Property Graphs and classical Hypergraphs. In applications where hyperedges of varying cardinalities carry labels and attributes, Property HyperGraphs offer a unified modelling framework.

3 Main Result: Property SuperHyperGraphs

We introduce and formalise the concept of a Property SuperHyperGraph.

Fix a finite base set V_0 and three (possibly infinite) sets

$$\Sigma \quad (n\text{-hyperedge–label alphabet}), \qquad K \quad (\text{property keys}), \qquad S \quad (\text{property values}),$$

together with a distinguished symbol $\perp \notin S$.

Definition 3.1 (Property n-SuperHyperGraph)**.** Define iterated powersets

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

A *property n-SuperHyperGraph* is a quadruple

$$H^{(n)} = \big(V^{(n)}, E^{(n)}, \lambda, \mu\big)$$

where

(a) $V^{(n)} \subseteq \mathcal{P}^n(V_0)$ and $E^{(n)}$ is a finite family of non-empty subsets of $\mathcal{P}^n(V_0)$.

(b) $\lambda : E^{(n)} \to \Sigma$ assigns a label to each n-superedge.

(c) $\mu : (V^{(n)} \cup E^{(n)}) \times K \to S \cup \{\perp\}$ is the property map, with

$$\mu(x,k) = \begin{cases} s \in S, & \text{if } x \text{ has key } k \text{ with value } s, \\ \perp, & \text{otherwise.} \end{cases}$$

We write

$$\operatorname{keyset}(x) = \{\, k \in K \mid \mu(x,k) \neq \perp \}, \quad \operatorname{val}(x,k) = \mu(x,k) \quad (k \in \operatorname{keyset}(x)).$$

Example 3.2 (Property 3-SuperHyperGraph for Image Classification Dataset)**.** We model a hierarchical image classification dataset with individual images (level 0), species classes (level 1), genera groups (level 2), and dataset splits (level 3).

Base set of images (V_0).

$$V_0 = \{\, i_1, i_2, i_3, i_4, i_5, i_6 \}.$$

Assign each image properties:

$$\mu(i_j, \texttt{resolution}) = \text{``1024×768''}, \quad \mu(i_j, \texttt{date}) = \text{``2025-06-01''} \quad (j = 1, \ldots, 6).$$

Species classes ($\mathcal{P}^1(V_0)$: 1-supervertices). Group images by species:

$$C_{\text{Cat}} = \{i_1, i_2\}, \quad C_{\text{Dog}} = \{i_3, i_4\}, \quad C_{\text{Bird}} = \{i_5, i_6\}.$$

Set

$$V^{(1)} = \{C_{\text{Cat}}, C_{\text{Dog}}, C_{\text{Bird}}\},$$

with properties

$$\mu(C_{\text{Cat}}, \texttt{species}) = \text{``Felis catus''}, \quad \mu(C_{\text{Cat}}, \texttt{count}) = 2,$$
$$\mu(C_{\text{Dog}}, \texttt{species}) = \text{``Canis lupus''}, \quad \mu(C_{\text{Dog}}, \texttt{count}) = 2,$$
$$\mu(C_{\text{Bird}}, \texttt{species}) = \text{``Passer domesticus''}, \quad \mu(C_{\text{Bird}}, \texttt{count}) = 2.$$

Genus groups ($\mathcal{P}^2(V_0)$: 2-supervertices). Group species by genus:

$$G_{\text{Felidae}} = \{C_{\text{Cat}}\}, \quad G_{\text{Canidae}} = \{C_{\text{Dog}}\}, \quad G_{\text{Passeridae}} = \{C_{\text{Bird}}\}.$$

Set

$$V^{(2)} = \{G_{\text{Felidae}}, G_{\text{Canidae}}, G_{\text{Passeridae}}\},$$

with properties

$$\mu(G_{\text{Felidae}}, \texttt{genus}) = \text{``Felis''}, \quad \mu(G_{\text{Felidae}}, \texttt{numSpecies}) = 1,$$
$$\mu(G_{\text{Canidae}}, \texttt{genus}) = \text{``Canis''}, \quad \mu(G_{\text{Canidae}}, \texttt{numSpecies}) = 1,$$
$$\mu(G_{\text{Passeridae}}, \texttt{genus}) = \text{``Passer''}, \quad \mu(G_{\text{Passeridae}}, \texttt{numSpecies}) = 1.$$

Dataset splits ($\mathcal{P}^3(V_0)$: 3-supervertices). Partition genera into splits:

$$S_{\text{Train}} = \{G_{\text{Felidae}}, G_{\text{Canidae}}\}, \quad S_{\text{Test}} = \{G_{\text{Passeridae}}\}.$$

Set

$$V^{(3)} = \{S_{\text{Train}}, S_{\text{Test}}\},$$

with properties

$$\mu(S_{\text{Train}}, \texttt{split}) = \text{``Train''}, \quad \mu(S_{\text{Train}}, \texttt{fraction}) = 0.8,$$
$$\mu(S_{\text{Test}}, \texttt{split}) = \text{``Test''}, \quad \mu(S_{\text{Test}}, \texttt{fraction}) = 0.2.$$

3-Superedges (full dataset). Link both splits into the complete dataset:

$$E^{(3)} = \{ e_{\text{Full}} \}, \quad e_{\text{Full}} = \{S_{\text{Train}}, S_{\text{Test}}\}.$$

Let the label alphabet be $\Sigma = \{\texttt{ImageNetSubset}\}$. Then

$$\lambda(e_{\text{Full}}) = \texttt{ImageNetSubset}, \quad \mu(e_{\text{Full}}, \texttt{version}) = \text{``v1.0''}.$$

Keysets and values. For example,

$$\text{keyset}(C_{\text{Dog}}) = \{\texttt{species}, \texttt{count}\}, \quad \text{val}(C_{\text{Dog}}, \texttt{species}) = \text{``Canis lupus''},$$

$$\text{keyset}(G_{\text{Passeridae}}) = \{\texttt{genus}, \texttt{numSpecies}\}, \quad \text{val}(G_{\text{Passeridae}}, \texttt{numSpecies}) = 1,$$

$$\text{keyset}(e_{\text{Full}}) = \{\texttt{version}\}, \quad \text{val}(e_{\text{Full}}, \texttt{version}) = \text{``v1.0''}.$$

Thus

$$H^{(3)} = \left(V^{(3)}, E^{(3)}, \lambda, \mu\right)$$

satisfies Definition 3.1, modeling a hierarchical image-classification dataset with richly attributed images, species classes, genera, and dataset splits.

Example 3.3 (Property 3-SuperHyperGraph for Multi-Tier Database Systems)**.** We model a typical multi-tier database architecture with columns (level 0), tables (level 1), schemas (level 2), and clusters of schemas (level 3).

Level 0 — Columns (V_0).

$$V_0 = \{ \mathit{user\_id}, \mathit{username}, \mathit{email}, \mathit{order\_id}, \mathit{order\_date}, \mathit{product\_id}, \mathit{quantity}, \mathit{price} \}.$$

Assign each column properties:

$$\mu(c, \texttt{name}) = c, \quad \mu(c, \texttt{dataType}) \in \{\texttt{INT}, \texttt{VARCHAR}, \texttt{DATE}, \texttt{DECIMAL}\}.$$

Level 1 — Tables ($\mathcal{P}^1(V_0)$). Define three tables as sets of columns:

$$T_{\mathsf{Users}} = \{\mathit{user\_id}, \mathit{username}, \mathit{email}\},$$

$$T_{\mathsf{Orders}} = \{\mathit{order\_id}, \mathit{user\_id}, \mathit{order\_date}\},$$

$$T_{\mathsf{Items}} = \{\mathit{order\_id}, \mathit{product\_id}, \mathit{quantity}, \mathit{price}\}.$$

Set

$$V^{(1)} = \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{Items}}\},$$

and assign

$$\mu(T_{\mathsf{Users}}, \texttt{tableName}) = \text{``Users''}, \quad \mu(T_{\mathsf{Users}}, \texttt{rowCount}) = 120000,$$

$$\mu(T_{\mathsf{Orders}}, \texttt{tableName}) = \text{``Orders''}, \quad \mu(T_{\mathsf{Orders}}, \texttt{rowCount}) = 450000,$$

$$\mu(T_{\mathsf{Items}}, \texttt{tableName}) = \text{``Items''}, \quad \mu(T_{\mathsf{Items}}, \texttt{rowCount}) = 950000.$$

Level 2 — Schemas ($\mathcal{P}^2(V_0)$). Group tables into logical schemas:

$$S_{\text{Sales}} = \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{Items}}\}, \quad V^{(2)} = \{S_{\text{Sales}}\}.$$

Assign schema properties:

$$\mu(S_{\text{Sales}}, \texttt{schemaName}) = \text{``SalesDB''}, \quad \mu(S_{\text{Sales}}, \texttt{version}) = \text{``v1.2''}.$$

Level 3 — Clusters ($\mathcal{P}^3(V_0)$). Define two clusters of schemas for high availability:

$$C_{\text{Primary}} = \{S_{\text{Sales}}\}, \quad C_{\text{Replica}} = \{S_{\text{Sales}}\}, \quad V^{(3)} = \{C_{\text{Primary}}, C_{\text{Replica}}\}.$$

Assign cluster properties:

$$\mu(C_{\text{Primary}}, \texttt{clusterRole}) = \text{``Primary''}, \quad \mu(C_{\text{Primary}}, \texttt{region}) = \text{``us-east-1''},$$

$$\mu(C_{\text{Replica}}, \texttt{clusterRole}) = \text{``Replica''}, \quad \mu(C_{\text{Replica}}, \texttt{region}) = \text{``us-west-2''}.$$

3-Superedges — Enterprise System. Link both clusters under the enterprise system:

$$E^{(3)} = \{ e_{\text{Enterprise}} \}, \qquad e_{\text{Enterprise}} = \{C_{\text{Primary}}, C_{\text{Replica}}\}.$$

Let $\Sigma = \{\texttt{EnterpriseDB}\}$. Then

$$\lambda(e_{\text{Enterprise}}) = \texttt{EnterpriseDB}, \quad \mu(e_{\text{Enterprise}}, \texttt{admin}) = \text{``DBA Team''}, \quad \mu(e_{\text{Enterprise}}, \texttt{uptime\_SLA}) = 99.99.$$

Keysets and Values. For example,

$$\text{keyset}(T_{\text{Items}}) = \{\texttt{tableName}, \texttt{rowCount}\}, \quad \text{val}(T_{\text{Items}}, \texttt{rowCount}) = 950000,$$

$$\text{keyset}(C_{\text{Replica}}) = \{\texttt{clusterRole}, \texttt{region}\}, \quad \text{val}(C_{\text{Replica}}, \texttt{region}) = \text{``us-west-2''},$$

$$\text{keyset}(e_{\text{Enterprise}}) = \{\texttt{admin}, \texttt{uptime\_SLA}\}, \quad \text{val}(e_{\text{Enterprise}}, \texttt{uptime\_SLA}) = 99.99.$$

Thus

$$H^{(3)} = \left(V^{(3)}, E^{(3)}, \lambda, \mu\right)$$

satisfies Definition 3.1, modeling a distributed multi-tier database system with richly attributed columns, tables, schemas, and clusters.

Example 3.4 (Property 2-SuperHyperGraph for Corporate Organization)**.** We model a company's hierarchy with employees (level 0), teams (level 1), and departments (level 2).

Base set of employees (V_0).

$$V_0 = \{\text{ Alice, Bob, Carol, Dave}\}.$$

Each employee $v \in V_0$ carries properties:

$$\mu(v, \texttt{name}) = \begin{cases} \text{``Alice''}, & v = \text{Alice}, \\ \text{``Bob''}, & v = \text{Bob}, \\ \text{``Carol''}, & v = \text{Carol}, \\ \text{``Dave''}, & v = \text{Dave}, \end{cases} \quad \mu(v, \texttt{role}) \in \{\text{``Engineer''}, \text{``Manager''}\},$$

$$\mu(v, \texttt{hireYear}) \in \{2018, 2019, 2020, 2021\}.$$

Teams ($\mathcal{P}^1(V_0)$: 1-supervertices). Define two teams:

$$T_A = \{\text{Alice}, \text{Bob}\}, \quad T_B = \{\text{Carol}, \text{Dave}\}.$$

Set

$$V^{(1)} = \{T_A, T_B\}, \quad \mu(T_A, \texttt{teamLead}) = \text{``Alice''}, \quad \mu(T_A, \texttt{location}) = \text{``HQ–Building 1''},$$

$$\mu(T_B, \texttt{teamLead}) = \text{``Carol''}, \quad \mu(T_B, \texttt{location}) = \text{``Office 2''}.$$

Departments ($\mathcal{P}^2(V_0)$: 2-supervertices). Form two departments as sets of teams:

$$D_X = \{T_A\}, \quad D_Y = \{T_B\}, \quad V^{(2)} = \{D_X, D_Y\}.$$

Assign

$$\mu(D_X, \texttt{deptHead}) = \text{``Bob''}, \quad \mu(D_X, \texttt{budget}) = 500000,$$
$$\mu(D_Y, \texttt{deptHead}) = \text{``Dave''}, \quad \mu(D_Y, \texttt{budget}) = 300000.$$

2-Superedges (linking departments). We group both departments under the same company:

$$E^{(2)} = \left\{ e_{\text{Corp}} \right\}, \quad e_{\text{Corp}} = \{D_X, D_Y\}.$$

Let the label alphabet $\Sigma = \{\texttt{AcmeCorp}\}$. Then

$$\lambda(e_{\text{Corp}}) = \texttt{AcmeCorp}, \quad \mu(e_{\text{Corp}}, \texttt{fiscalYear}) = 2025, \quad \mu(e_{\text{Corp}}, \texttt{region}) = \text{``Global''}.$$

Keysets and values. For instance,

$$\text{keyset(Carol)} = \{\texttt{name}, \texttt{role}, \texttt{hireYear}\}, \quad \text{val(Carol}, \texttt{role}) = \text{``Manager''},$$

$$\text{keyset}(D_X) = \{\texttt{deptHead}, \texttt{budget}\}, \quad \text{val}(D_X, \texttt{budget}) = 500000,$$
$$\text{keyset}(e_{\text{Corp}}) = \{\texttt{fiscalYear}, \texttt{region}\}, \quad \text{val}(e_{\text{Corp}}, \texttt{region}) = \text{``Global''}.$$

Thus $H^{(2)} = (V^{(2)}, E^{(2)}, \lambda, \mu)$ satisfies Definition 3.1: supervertices are departments, superedges link departments into the company, and each entity carries labels and properties describing roles, budgets, and organizational structure.

Example 3.5 (Property 3-SuperHyperGraph for Environmental Monitoring Network)**.** We model a three-level environmental monitoring network: individual sensors (level 0), sensor clusters (level 1), regions (level 2), and national zones (level 3).

Base set of sensors (V_0).

$$V_0 = \{\, s_1,\ s_2,\ s_3,\ s_4\}.$$

Assign each sensor two properties:

$$\mu(s_1, \texttt{location}) = \text{``North Field''}, \quad \mu(s_1, \texttt{sensorType}) = \text{``Temperature''},$$

$$\mu(s_2, \texttt{location}) = \text{``North Field''}, \quad \mu(s_2, \texttt{sensorType}) = \text{``Humidity''},$$
$$\mu(s_3, \texttt{location}) = \text{``South Valley''}, \quad \mu(s_3, \texttt{sensorType}) = \text{``Temperature''},$$
$$\mu(s_4, \texttt{location}) = \text{``South Valley''}, \quad \mu(s_4, \texttt{sensorType}) = \text{``Humidity''}.$$

Sensor clusters ($\mathcal{P}^1(V_0)$: 1-supervertices). Form two clusters of nearby sensors:

$$C_N = \{s_1, s_2\}, \quad C_S = \{s_3, s_4\}, \quad V^{(1)} = \{C_N, C_S\}.$$

Each cluster carries properties:

$$\mu(C_N, \texttt{clusterID}) = \text{``North-01''}, \quad \mu(C_N, \texttt{manager}) = \text{``Alice''},$$

$$\mu(C_S, \texttt{clusterID}) = \text{``South-01''}, \quad \mu(C_S, \texttt{manager}) = \text{``Bob''}.$$

Regions ($\mathcal{P}^2(V_0)$: 2-supervertices). Group clusters into two regions:

$$R_{\text{North}} = \{C_N\}, \quad R_{\text{South}} = \{C_S\}, \quad V^{(2)} = \{R_{\text{North}}, R_{\text{South}}\}.$$

Assign regional properties:

$$\mu(R_{\text{North}}, \texttt{regionName}) = \text{``Northern Zone''}, \quad \mu(R_{\text{North}}, \texttt{supervisor}) = \text{``Carol''},$$

$$\mu(R_{\text{South}}, \texttt{regionName}) = \text{``Southern Zone''}, \quad \mu(R_{\text{South}}, \texttt{supervisor}) = \text{``Dave''}.$$

Zones ($\mathcal{P}^3(V_0)$: 3-supervertices). Define two national zones:

$$Z_{\text{A}} = \{R_{\text{North}}\}, \quad Z_{\text{B}} = \{R_{\text{South}}\}, \quad V^{(3)} = \{Z_{\text{A}}, Z_{\text{B}}\}.$$

Each zone has properties:

$$\mu(Z_{\text{A}}, \texttt{zoneCode}) = \text{``ZA''}, \quad \mu(Z_{\text{A}}, \texttt{climate}) = \text{``Temperate''},$$

$$\mu(Z_{\text{B}}, \texttt{zoneCode}) = \text{``ZB''}, \quad \mu(Z_{\text{B}}, \texttt{climate}) = \text{``Arid''}.$$

3-superedges (national network). We link both zones under the national monitoring system:

$$E^{(3)} = \{e_{\text{Nat}}\}, \quad e_{\text{Nat}} = \{Z_{\text{A}}, Z_{\text{B}}\}.$$

Let the label alphabet be $\Sigma = \{\texttt{EnvNet}\}$. Then

$$\lambda(e_{\text{Nat}}) = \texttt{EnvNet}, \quad \mu(e_{\text{Nat}}, \texttt{operator}) = \text{``EnvDept''}, \quad \mu(e_{\text{Nat}}, \texttt{capacity}) = 10000.$$

Keysets and values. For example,

$$\text{keyset}(s_1) = \{\texttt{location}, \texttt{sensorType}\}, \quad \text{val}(s_1, \texttt{sensorType}) = \text{``Temperature''},$$

$$\text{keyset}(R_{\text{South}}) = \{\texttt{regionName}, \texttt{supervisor}\}, \quad \text{val}(R_{\text{South}}, \texttt{supervisor}) = \text{``Dave''},$$
$$\text{keyset}(e_{\text{Nat}}) = \{\texttt{operator}, \texttt{capacity}\}, \quad \text{val}(e_{\text{Nat}}, \texttt{capacity}) = 10000.$$

Thus

$$H^{(3)} = \left(V^{(3)}, E^{(3)}, \lambda, \mu\right)$$

satisfies Definition 3.1, modelling a hierarchical environmental-sensor network with richly attributed sensors, clusters, regions, and zones.

Theorem 3.6 (Generalisation of Property Graphs, HyperGraphs, and n-SHGs)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph as in Definition 3.1. Then:

(i) If $n = 1$, every 1-superedge has size two, and we choose an orientation on each, then $H^{(1)}$ reduces to a Property Graph as in Definition 1.10.

(ii) If $n = 1$, $\Sigma = \{\sigma_0\}$ is a singleton, and $\mu \equiv \bot$, then $H^{(1)}$ collapses to a classical Hypergraph $(V^{(1)}, E^{(1)})$.

(iii) If $\Sigma = \{\sigma_0\}$ and $\mu \equiv \bot$, then $H^{(n)}$ collapses to an ordinary n-SuperHyperGraph $(V^{(n)}, E^{(n)})$.

Proof. (i) When $n = 1$, we have $V^{(1)} \subseteq \mathcal{P}(V_0)$ and $E^{(1)} \subseteq \{\text{non-empty subsets of } V_0\}$. Requiring each hyperedge $e \in E^{(1)}$ to satisfy $|e| = 2$ and then choosing an ordering (u, v) endows e with source and target functions $s(e) = u$, $t(e) = v$. The maps λ and μ then satisfy exactly the axioms of a Property Graph.

(ii) If Σ is a singleton and $\mu \equiv \perp$, then labels and properties carry no information. Thus $H^{(1)} = (V^{(1)}, E^{(1)})$ satisfies precisely the definition of a Hypergraph.

(iii) For general n, the same argument shows that trivializing labels and properties makes $H^{(n)}$ an ordinary n-SuperHyperGraph, since only the sets $V^{(n)}$ and $E^{(n)}$ remain. □

Remark 3.7. Theorem 3.6 demonstrates that Property n-SuperHyperGraphs form a single, coherent framework encompassing:

- Property Graphs (as $n = 1$ with 2-uniform edges),
- classical HyperGraphs (as trivial properties and labels, $n = 1$),
- ordinary n-SuperHyperGraphs (as trivial properties and labels).

This unification facilitates the modelling of richly-attributed, multi-layered relational data.

Theorem 3.8 (Induced Sub-n-SuperHyperGraph)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph on the base set V_0. For any subset $U_0 \subseteq V_0$ define

$$V_U^{(n)} := \{x \in V^{(n)} \mid x \subseteq \mathcal{P}^{n-1}(U_0)\}, \quad E_U^{(n)} := \{e \in E^{(n)} \mid e \subseteq V_U^{(n)}\}.$$

Equip $H_U^{(n)} = (V_U^{(n)}, E_U^{(n)}, \lambda|_{E_U^{(n)}}, \mu|_{(V_U^{(n)} \cup E_U^{(n)}) \times K})$. Then $H_U^{(n)}$ is again a property n-SuperHyperGraph (on base U_0).

Proof. We must check the three parts of Definition 3.1 for $H_U^{(n)}$:

(a) By construction $V_U^{(n)} \subseteq \mathcal{P}^n(U_0) \subseteq \mathcal{P}^n(V_0)$ and $E_U^{(n)}$ is a family of non-empty subsets of $V_U^{(n)} \subseteq \mathcal{P}^n(U_0)$.

(b) The label map λ restricted to $E_U^{(n)}$ still takes values in Σ.

(c) The property map μ, when restricted to $(V_U^{(n)} \cup E_U^{(n)}) \times K$, still assigns to each pair either an element of S or $\perp$.

All axioms of Definition 3.1 are therefore inherited, so $H_U^{(n)}$ is a property n-SuperHyperGraph on the smaller base U_0. □

Theorem 3.9 (Flattening Projection)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph. Define the flattening map

$$\varphi\colon \mathcal{P}^n(V_0) \longrightarrow \mathcal{P}^{n-1}(V_0), \quad \varphi(X) = \bigcup_{x \in X} x.$$

Assume that

- *λ is* constant *on each fibre of φ (so that λ factors through φ), and*
- *$\mu(y, k) = \mu(y', k)$ whenever $\varphi(y) = \varphi(y')$ for all $k \in K$.*

Then

$$H' \;=\; \big(\varphi(V^{(n)}),\ \varphi(E^{(n)}),\ \lambda',\ \mu'\big)$$

is a property $(n-1)$-SuperHyperGraph, where

$$\lambda'(e') \;=\; \lambda(e) \quad (\varphi(e) = e'), \qquad \mu'(z', k) \;=\; \mu(z, k) \quad (\varphi(z) = z').$$

Proof. We verify Definition 3.1 for H' with "n" replaced by "$n-1$":

(a) By definition $\varphi(V^{(n)}) \subseteq \mathcal{P}^{n-1}(V_0)$ and each $\varphi(e) \subseteq \varphi(V^{(n)})$ is non-empty.

(b) λ' is well-defined because λ is constant on fibres. Its codomain remains Σ.

(c) μ' is well-defined by the second assumption; it maps into $S \cup \{\perp\}$.

Hence H' satisfies all requirements of a property $(n-1)$-SuperHyperGraph. □

Theorem 3.10 (Uniformity and k-Uniform Level-n Substructures)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph. Suppose there exists an integer $k \geq 1$ such that for every $e \in E^{(n)}$, $|e| = k$. Then $(V^{(n)}, E^{(n)}, \lambda, \mu)$ is a k-uniform property n-SuperHyperGraph*, and in particular each level-n superedge links exactly k supervertices. Moreover, any subset $E'^{(n)} \subseteq E^{(n)}$ inherits the same uniformity.*

Proof. By hypothesis, every n-superedge $e \in E^{(n)}$ has cardinality k, so the definition of k-uniformity is immediate. Restricting to $E'^{(n)}$ clearly preserves the condition $|e| = k$ for all $e \in E'^{(n)}$. Labels and properties remain unchanged under restriction, so all axioms of Definition 3.1 hold on the uniform subfamily. □

Theorem 3.11 (Disjoint Union)**.** *Let*

$$H_1^{(n)} = (V_1^{(n)}, E_1^{(n)}, \lambda_1, \mu_1), \quad H_2^{(n)} = (V_2^{(n)}, E_2^{(n)}, \lambda_2, \mu_2)$$

be two property n-SuperHyperGraphs over disjoint base sets $V_0^{(1)}$ and $V_0^{(2)}$, with $V_0^{(1)} \cap V_0^{(2)} = \varnothing$. Define

$$V^{(n)} \;=\; V_1^{(n)} \;\cup\; V_2^{(n)}, \quad E^{(n)} \;=\; E_1^{(n)} \;\cup\; E_2^{(n)},$$

and extend λ and μ by

$$\lambda(e) = \begin{cases} \lambda_1(e), & e \in E_1^{(n)}, \\ \lambda_2(e), & e \in E_2^{(n)}, \end{cases} \qquad \mu(x, k) = \begin{cases} \mu_1(x, k), & x \in V_1^{(n)} \cup E_1^{(n)}, \\ \mu_2(x, k), & x \in V_2^{(n)} \cup E_2^{(n)}. \end{cases}$$

Then $\big(V^{(n)}, E^{(n)}, \lambda, \mu\big)$ is a property n-SuperHyperGraph on base $V_0^{(1)} \cup V_0^{(2)}$.

Proof. We check Definition 3.1 for the union:

(a) Since $V_1^{(n)} \subseteq \mathcal{P}^n(V_0^{(1)})$ and $V_2^{(n)} \subseteq \mathcal{P}^n(V_0^{(2)})$, their union lies in $\mathcal{P}^n(V_0^{(1)} \cup V_0^{(2)})$. Likewise each $E_i^{(n)}$ is a family of non-empty subsets of $V_i^{(n)}$, so $E^{(n)}$ is a family of non-empty subsets of $V^{(n)}$.

(b) The label map λ is well-defined on $E^{(n)}$ and takes values in Σ.

(c) The property map μ is well-defined on $(V^{(n)} \cup E^{(n)}) \times K$, assigning each pair either an element of S or $\perp$.

All axioms of Definition 3.1 are therefore satisfied. □

Theorem 3.12 (Line Graph as Property Graph)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph. Its line graph $L(H^{(n)})$ *is defined as the septuple*

$$L(H^{(n)}) \;=\; \big(V_G,\, E_G,\, s,\, t,\, \lambda_G,\, \mu_G,\, \bot\big),$$

where

$$V_G = E^{(n)}, \quad E_G = \big\{\, (e_1, e_2) \mid e_1, e_2 \in E^{(n)},\ e_1 \neq e_2,\ e_1 \cap e_2 \neq \varnothing \big\},$$
$$s(e_1, e_2) = e_1, \quad t(e_1, e_2) = e_2,$$
$$\lambda_G(e_1, e_2) = \big(\lambda(e_1), \lambda(e_2)\big) \in \Sigma \times \Sigma, \quad \mu_G(x, k) = \begin{cases} \mu(x, k), & x \in V_G \cup E_G, \\ \bot, & \textit{otherwise,} \end{cases}$$

and $\bot \notin S$*. Then* $L(H^{(n)})$ *is a Property Graph in the sense of Definition 1.10.*

Proof. We check the clauses of Definition 1.10:

(a) $V_G = E^{(n)}$ is finite or countable since $E^{(n)}$ is finite.

(b) E_G is a finite set of ordered pairs of elements of V_G.

(c) The source and target functions $s, t : E_G \to V_G$ are given by projection.

(d) The label function $\lambda_G : E_G \to \Sigma \times \Sigma$ is well-defined.

(e) The property map $\mu_G : (V_G \cup E_G) \times K \to S \cup \{\bot\}$ extends the original μ and yields either a value in S or $\bot$.

(f) $\bot$ is not in S by construction.

Hence $L(H^{(n)})$ satisfies all requirements of a Property Graph. □

Theorem 3.13 (Iterated Flattening)**.** *Let*

$$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$$

be a property n-SuperHyperGraph and fix an integer $1 \leq k \leq n$*. Define the k-fold flattening*

$$\varphi^{(k)} \;=\; \underbrace{\varphi \circ \varphi \circ \cdots \circ \varphi}_{k \text{ times}} \;:\; \mathcal{P}^n(V_0) \;\longrightarrow\; \mathcal{P}^{n-k}(V_0),$$

where φ *is as in Theorem 3.9. If* λ *and* μ *factor through each intermediate* φ*, then*

$$H^{(n-k)} = \Big(\varphi^{(k)}(V^{(n)}),\, \varphi^{(k)}(E^{(n)}),\, \lambda^{(k)},\, \mu^{(k)}\Big)$$

is a property $(n-k)$*-SuperHyperGraph, with labels and properties induced by composition.*

Proof. We proceed by induction on k.

Base case ($k = 1$): This is exactly Theorem 3.9.

Induction step: Assume the statement holds for k. Then we have

$$H^{(n-k)} = \big(V^{(n-k)}, E^{(n-k)}, \lambda^{(k)}, \mu^{(k)}\big)$$

a property $(n-k)$-SHG. Applying Theorem 3.9 once more to $H^{(n-k)}$ yields a property $(n-k-1)$-SHG $\big(\varphi(V^{(n-k)}), \varphi(E^{(n-k)}), \lambda^{(k+1)}, \mu^{(k+1)}\big)$, where $\lambda^{(k+1)}$ and $\mu^{(k+1)}$ remain well-defined because they factor through φ. But $\varphi \circ \varphi^{(k)} = \varphi^{(k+1)}$, completing the induction. □

4 Conclusion and Outlook

This study has shown how the descriptive capabilities of Property Graphs can be elevated to the richer frameworks of HyperGraphs and SuperHyperGraphs. We provided rigorous definitions for each generalisation and reported initial results that highlight their modelling potential.

Future work may explore further extensions based on advanced uncertainty formalisms—including Fuzzy Sets [39, 40], Vague Sets [41, 42], Intuitionistic Fuzzy Sets [43, 44], Paraconsistent Set [45–47], Soft Sets [48, 49], Picture Fuzzy Set [50, 51], Rough Sets [52–54], Neutrosophic Sets (and their quadri-partitioned variants) [55–57], HyperFuzzy Sets [58–62], Hesitant Fuzzy Sets [63, 64], and Plithogenic Sets [65, 66]. Each of these set-theoretic paradigms already possesses a graph-theoretic interpretation, and incorporating them into the Property HyperGraph and Property SuperHyperGraph frameworks promises a fertile direction for research.

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.

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.

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] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[2] Jonathan L Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.

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

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

[5] Mohammad Hamidi, Florentin Smarandache, and Mohadeseh Taghinezhad. *Decision Making Based on Valued Fuzzy Superhypergraphs*. Infinite Study, 2023.

[6] Oskar Van Rest, Sungpack Hong, Jinha Kim, Xuming Meng, and Hassan Chafi. Pgql: a property graph query language. In *Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems*, pages 1–6, 2016.

[7] Wen Sun, Achille Fokoue, Kavitha Srinivas, Anastasios Kementsietsidis, Gang Hu, and Guotong Xie. Sqlgraph: An efficient relational-based property graph store. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1887–1901, 2015.

[8] Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, et al. Pg-schema: Schemas for property graphs. *Proceedings of the ACM on Management of Data*, 1(2):1–25, 2023.

[9] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2020.

[10] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[11] Bibin K Jose and Zsolt Tuza. Hypergraph domination and strong independence. *Applicable Analysis and Discrete Mathematics*, 3(2):347–358, 2009.

[12] Muhammad Akram and Gulfam Shahzadi. Hypergraphs in m-polar fuzzy environment. *Mathematics*, 6(2):28, 2018.

[13] Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, Shenda Hong, and Hongyan Li. Hypergraph structure learning for hypergraph neural networks. In *IJCAI*, pages 1923–1929, 2022.

[14] 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.

[15] Takaaki Fujita. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*. Biblio Publishing, 2025.

[16] Mohammed Alqahtani. Intuitionistic fuzzy quasi-supergraph integration for social network decision making. *International Journal of Analysis and Applications*, 23:137–137, 2025.

[17] N. B. Nalawade, M. S. Bapat, S. G. Jakkewad, G. A. Dhanorkar, and D. J. Bhosale. Structural properties of zero-divisor hypergraph and superhypergraph over $\mathbb{Z}_n$: Girth and helly property. *Panamerican Mathematical Journal*, 35(4S):485, 2025.

[18] Takaaki Fujita. Hypergraph and superhypergraph approaches in electronics: A hierarchical framework for modeling power-grid hypernetworks and superhypernetworks. *Journal of Energy Research and Reviews*, 17(6):102–136, 2025.

[19] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

[20] Takaaki Fujita. An introduction and reexamination of molecular hypergraph and molecular n-superhypergraph. *Asian Journal of Physical and Chemical Sciences*, 13(3):1–38, 2025.

[21] Berrocal Villegas Salomón Marcos, Montalvo Fritas Willner, Berrocal Villegas Carmen Rosa, Flores Fuentes Rivera María Yissel, Espejo Rivera Roberto, Laura Daysi Bautista Puma, and Dante Manuel Macazana Fernández. Using plithogenic n-superhypergraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems*, 84:513–524, 2025.

[22] Shouxian Zhu. Neutrosophic n-superhypernetwork: A new approach for evaluating short video communication effectiveness in media convergence. *Neutrosophic Sets and Systems*, 85:1004–1017, 2025.

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

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

[25] Huda E Khali, Gonca D GÜNGÖR, and Muslim A Noah Zaina. Neutrosophic superhyper bi-topological spaces: Original notions and new insights. *Neutrosophic Sets and Systems*, 51(1):3, 2022.

[26] 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.

[27] S Santhakumar, IR Sumathi, and J Mahalakshmi. A novel approach to the algebraic structure of neutrosophic superhyper algebra. *Neutrosophic Sets and Systems*, 60(1):39, 2023.

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

[29] 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.

[30] Florentin Smarandache. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*. Infinite Study, 2020.

[31] Takaaki Fujita and Arkan A Ghaib. Toward a unified theory of brain hypergraphs and symptom hypernetworks in medicine and neuroscience. *Advances in Research*, 26(3):522–565, 2025.

[32] Eldar Fischer and Ilan Newman. Testing versus estimation of graph properties. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 138–146, 2005.

[33] Pietro Parodi and Peter Watson. Property graphs–a statistical model for fire and explosion losses based on graph theory. *ASTIN Bulletin: The Journal of the IAA*, 49(2):263–297, 2019.

[34] Soonbo Han and Zachary G Ives. Implementation strategies for views over property graphs. *Proceedings of the ACM on Management of Data*, 2(3):1–26, 2024.

[35] Łukasz Szeremeta, Dominik Tomaszuk, and Renzo Angles. Yars-pg: Property graphs representation for publication and exchange. *IEEE Access*, 2024.

[36] Souripriya Das, Jagannathan Srinivasan, Matthew Perry, Eugene Inseok Chong, and Jayanta Banerjee. A tale of two graphs: Property graphs as rdf in oracle. In *EDBT*, pages 762–773, 2014.

[37] Soonbo Han and Zachary G Ives. Implementing views for property graphs. *ACM SIGMOD Record*, 54(1):59–68, 2025.

[38] Philipp Skavantzos and Sebastian Link. Normalizing property graphs. *Proceedings of the VLDB Endowment*, 16(11):3031–3043, 2023.

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

[40] Lotfi A Zadeh. A note on prototype theory and fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: Selected papers by Lotfi A Zadeh*, pages 587–593. World Scientific, 1996.

[41] W-L Gau and Daniel J Buehrer. Vague sets. *IEEE transactions on systems, man, and cybernetics*, 23(2):610–614, 1993.

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

[43] Krassimir T Atanassov and Krassimir T Atanassov. *Intuitionistic fuzzy sets*. Springer, 1999.

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

[45] Walter Alexandre Carnielli and Marcelo Esteban Coniglio. Paraconsistent set theory by predicating on consistency. *J. Log. Comput.*, 26:97–116, 2016.

[46] Aida Vitória, Jan Maluszynski, and Andrzej Sza?as. Modeling and reasoning with paraconsistent rough sets. *Fundam. Informaticae*, 97:405–438, 2009.

[47] Zach Weber. Transfinite numbers in paraconsistent set theory. *The Review of Symbolic Logic*, 3:71 – 92, 2010.

[48] Dmitriy Molodtsov. Soft set theory-first results. *Computers & mathematics with applications*, 37(4-5):19–31, 1999.

[49] Pradip Kumar Maji, Ranjit Biswas, and A Ranjan Roy. Soft set theory. *Computers & mathematics with applications*, 45(4-5):555–562, 2003.

[50] Xiaolong Shi, Saeed Kosari, Ali Asghar Talebi, Seyed Hossein Sadati, and Hossein Rashmanlou. Investigation of the main energies of picture fuzzy graph and its applications. *International Journal of Computational Intelligence Systems*, 15(1):31, 2022.

[51] 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.

[52] Zdzisław Pawlak. Rough sets. *International journal of computer & information sciences*, 11:341–356, 1982.

[53] Xuerong Zhao and Bao Qing Hu. Fuzzy and interval-valued fuzzy decision-theoretic rough set approaches based on fuzzy probability measure. *Inf. Sci.*, 298:534–554, 2015.

[54] Zdzisław Pawlak. *Rough sets: Theoretical aspects of reasoning about data*, volume 9. Springer Science & Business Media, 2012.

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

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

[57] R Radha, A Stanis Arul Mary, and Florentin Smarandache. Quadripartitioned neutrosophic pythagorean soft set. *International Journal of Neutrosophic Science (IJNS) Volume 14, 2021*, page 11, 2021.

[58] Young Bae Jun, Kul Hur, and Kyoung Ja Lee. Hyperfuzzy subalgebras of bck/bci-algebras. *Annals of Fuzzy Mathematics and Informatics*, 2017.

[59] Takaaki Fujita and Florentin Smarandache. Examples of fuzzy sets, hyperfuzzy sets, and superhyperfuzzy sets in climate change and the proposal of several new concepts. *Climate Change Reports*, 2:1–18, 2025.

[60] Hashem Bordbar, Mohammad Rahim Bordbar, Rajab Ali Borzooei, and Young Bae Jun. N-subalgebras of bck= bci-algebras which are induced from hyperfuzzy structures. *Iranian Journal of Mathematical Sciences and Informatics*, 16(2):179–195, 2021.

[61] Takaaki Fujita and Florentin Smarandache. A concise introduction to hyperfuzzy, hyperneutrosophic, hyperplithogenic, hypersoft, and hyperrough sets with practical examples. *Neutrosophic Sets and Systems*, 80:609–631, 2025.

[62] Z Nazari and B Mosapour. The entropy of hyperfuzzy sets. *Journal of Dynamical Systems and Geometric Theories*, 16(2):173–185, 2018.

[63] Vicenç Torra and Yasuo Narukawa. On hesitant fuzzy sets and decision. In *2009 IEEE international conference on fuzzy systems*, pages 1378–1382. IEEE, 2009.

[64] Vicenç Torra. Hesitant fuzzy sets. *International journal of intelligent systems*, 25(6):529–539, 2010.

[65] Nivetha Martin. Plithogenic swara-topsis decision making on food processing methods with different normalization techniques. *Advances in Decision Making*, 69, 2022.

[66] 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.