EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 4, Article Number 6729
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Property HyperGraphs and Property SuperHyperGraphs for Data Analysis

Takaaki Fujita[1,*], Florentin Smarandache[2]

[1] *Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan*
[2] *University of New Mexico, Gallup Campus, NM 87301, USA*

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.

2020 Mathematics Subject Classifications: 05C65

Key Words and Phrases: Property graphs, superhypergraphs, hypergraphs, property superhypergraphs, property hypergraphs

1. Introduction

Classical graphs model binary relations by representing entities as vertices and their pairwise connections as edges [1, 9]. Hypergraphs extend this notion by allowing each hyperedge to join any nonempty collection of vertices, thereby capturing higher-order interactions [10–12]. However, even hypergraphs cannot naturally express nested or hierarchical groupings. To remedy this, *SuperHyperGraphs* were introduced: by iteratively applying the powerset operation to the vertex set, one obtains multi-layered networks that encode nested relationships among vertex clusters [13–15]. This construction is versatile enough to subsume a wide range of graph models—including HyperGraphs and MultiHyperGraphs—under a single unifying framework.

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i4.6729

Email addresses: `Takaaki.fujita060@gmail.com` (T. Fujita),
`smarand@unm.edu` (F. Smarandache)

Directed variants of these structures—such as Directed Graphs [16], Directed HyperGraphs [17–19], and Directed SuperHyperGraphs[20]—are also well-known and have been extensively studied in numerous research papers.

Data modeling is the process of defining and organizing data structures, relationships, and constraints to support storage, retrieval, and analysis [21–23]. A *Property Graph* is a directed multigraph in which both vertices and edges carry arbitrary key–value attributes, and each edge is labeled. This schema-flexible model enables the representation of rich relational data [6–8].

Although research on HyperGraphs, SuperHyperGraphs, and Property Graphs is well established, the specific combinations—*Property HyperGraphs* and *Property SuperHyperGraphs*—have received little attention. In this paper, we fill this gap by providing formal definitions for Property HyperGraphs and Property SuperHyperGraphs and by presenting preliminary theoretical results that demonstrate their expressive power. The author believes that one of the advantages of these frameworks is that Property Graphs can also be applied to more hierarchical concepts.

This subsection explains the structure of the paper. Section 2 provides an overview of SuperHyperGraphs, HyperGraphs, and Property Graphs. Section 3 examines Property HyperGraphs, discussing their applications and key characteristics. Section 4 investigates Property SuperHyperGraphs, presenting applications and structural properties. Section 5 offers concluding remarks and outlines possible directions for future research.

2. 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.

2.1. SuperHyperGraphs

A finite *hypergraph* generalizes the classical graph model by permitting *hyperedges* that connect any non-empty subset of vertices [10, 24, 25]. 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 [26, 27]. Such structures have demonstrated utility in areas ranging from molecular design and complex-network analysis to advanced signal-processing pipelines [28–30]. Unless stated otherwise, the integer n in $P_n(\cdot)$ or in an n-SuperHyperGraph is assumed to be non-negative.

Definition 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 2 (Powerset)**.** *(cf.[31, 32]) 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 3 (Hypergraph)**.** *[3, 33] 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 (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 3.*

Vertex set. *Let*

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

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:*

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

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 4 (n-th Powerset)**.** *[34, 35] 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 2 (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:*

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

$P_3(X) = \mathcal{P}\big(P_2(X)\big)$: 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:*

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

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 5 (n-SuperHyperGraph)**.** *(cf. [13]) Fix a finite, nonempty base set V_0 and define the iterated powerset by*

$$\mathcal{P}^0(V_0) := V_0, \qquad \mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \quad (k \in \mathbb{N}).$$

*For an integer $n \geq 0$, an n-*SuperHyperGraph on *V_0 is a pair*

$$\mathrm{SHG}^{(n)} = (V, E)$$

such that

$$V \subseteq \mathcal{P}^n(V_0) \quad \textit{and} \quad E \subseteq \mathcal{P}(V) \setminus \{\varnothing\}.$$

*Elements of V are called n-*supervertices *and elements of E are n-*superedges*. (In particular, each n-superedge is a nonempty subset of V.)*

Table 1 provides a concise overview of Graphs, Hypergraphs, and n-SuperHyperGraphs. In particular, SuperHyperGraphs are expected to offer an intuitive way to represent hierarchical network structures that commonly arise in real-world systems.

Table 1: Concise overview of Graph, Hypergraph, and n-SuperHyperGraph.

Model	Vertices	Edges	Notes
Graph	V (finite)	$E \subseteq V \times V$ or $E = \{\{u, v\}\}$	Each edge links exactly two vertices; captures pairwise relations.
Hypergraph	V (finite)	$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	Hyperedges connect arbitrary nonempty subsets of vertices.
n-SuperHyperGraph	$V \subseteq \mathcal{P}^n(V_0)$	$E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	Vertices lie at level n (iterated powerset); edges link level-n vertices in hierarchical groupings.

Several concrete examples of SuperHyperGraphs are presented below.

Example 3 (A 2-SuperHyperGraph for market-basket data mining)**.** *We show that a toy transaction dataset naturally forms a 2-SuperHyperGraph $\mathrm{SHG}^{(2)} = (V, E)$ in the sense of Definition 5.*

Step 0 (base set of items). *Let*

$$V_0 = \{A, B, C, D\} = \{\textit{bread}, \ \textit{milk}, \ \textit{eggs}, \ \textit{butter}\}.$$

Step 1 (transactions as 1-supervertices). *Transactions are subsets of V_0, hence elements of $\mathcal{P}^1(V_0) = \mathcal{P}(V_0)$. Fix four transactions*

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

Step 2 (pattern groups as 2-supervertices). *Sets of transactions lie in $\mathcal{P}^2(V_0) = \mathcal{P}\big(\mathcal{P}(V_0)\big)$. Define*

$$G_1 = \{T_1, T_4\}, \qquad G_2 = \{T_2, T_3\}, \qquad G_3 = \{T_2, T_3, T_4\},$$

and set

$$V := \{G_1, G_2, G_3\} \ \subseteq \ \mathcal{P}^2(V_0).$$

Interpretation: G_1 collects baskets with $\{A, B\}$; G_2 those with $\{C, D\}$; G_3 those with D plus at least one additional item.

Step 3 (2-superedges via overlap of pattern groups). *Define a symmetric adjacency on V by*

$$\{G_i, G_j\} \in E \quad \Longleftrightarrow \quad i \neq j \ \ \text{and} \ \ G_i \cap G_j \neq \varnothing,$$

and take

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

Explicitly, $G_1 \cap G_3 = \{T_4\} \neq \varnothing$ and $G_2 \cap G_3 = \{T_2, T_3\} \neq \varnothing$, while $G_1 \cap G_2 = \varnothing$.

Each $T_i \subseteq V_0 \Rightarrow T_i \in \mathcal{P}^1(V_0)$. Therefore $G_j \subseteq \{T_1, T_2, T_3, T_4\} \Rightarrow G_j \in \mathcal{P}^2(V_0)$. Thus $V \subseteq \mathcal{P}^2(V_0)$. Moreover, every listed superedge is a nonempty subset of V, so $E \subseteq \mathcal{P}(V) \setminus \{\varnothing\}$. Hence $\mathrm{SHG}^{(2)} = (V, E)$ is a valid 2-SuperHyperGraph.

Use in pattern mining. *Vertices G_j encode candidate frequent pattern groups; edges record higher-order overlaps among these groups. Level-wise algorithms (e.g. Apriori) may operate on $\mathrm{SHG}^{(2)}$ to prune infrequent groups and derive rules such as $\{A, B\} \Rightarrow D$ when supported by the underlying transactions (e.g. via the shared transaction T_4).*

Example 4 (A 3-SuperHyperGraph for multi-tier database development)**.** *We model columns, tables, logical databases, and services at successive powerset levels.*

Step 0 (V_0: atomic columns).

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

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

$$\left.\begin{aligned} T_{\mathsf{Users}} &:= \{\mathit{UserID}, \mathit{Name}, \mathit{Email}\}, \\ T_{\mathsf{Orders}} &:= \{\mathit{OrderID}, \mathit{UserID}, \mathit{OrderDate}\}, \\ T_{\mathsf{OrderItems}} &:= \{\mathit{OrderID}, \mathit{ProductID}, \mathit{Quantity}, \mathit{Price}\}, \end{aligned}\right. \qquad \in \ \mathcal{P}^1(V_0).$$

Let $V_1 := \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\} \subseteq \mathcal{P}^1(V_0)$. *Define 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 foreign-key relations Orders.UserID → Users.UserID and OrderItems.OrderID → Orders.OrderID.

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

$$\mathsf{EComDB} := \{T_{\mathsf{Users}}, T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\}, \qquad \mathsf{AnalyticsDB} := \{T_{\mathsf{Orders}}, T_{\mathsf{OrderItems}}\},$$

so $V_2 := \{\mathsf{EComDB}, \mathsf{AnalyticsDB}\} \subseteq \mathcal{P}^2(V_0)$. *Define a 2-superedge*

$$E_2 := \big\{\{\mathsf{EComDB}, \mathsf{AnalyticsDB}\}\big\} \ \subseteq\ \mathcal{P}(V_2) \setminus \{\varnothing\},$$

representing an ETL replication from operational to analytical storage.

Step 3 ($\mathcal{P}^3(V_0)$: services as 3-supervertices).

$$\mathsf{TransactionSvc} := \{\mathsf{EComDB}\}, \qquad \mathsf{ReportingSvc} := \{\mathsf{AnalyticsDB}\},$$

hence $V_3 := \{\mathsf{TransactionSvc}, \mathsf{ReportingSvc}\} \subseteq \mathcal{P}^3(V_0)$. *Define a 3-superedge*

$$E_3 := \big\{\{\mathsf{TransactionSvc}, \mathsf{ReportingSvc}\}\big\} \ \subseteq\ \mathcal{P}(V_3) \setminus \{\varnothing\},$$

capturing an event stream by which the transaction service publishes updates consumed by the reporting service.

Verification (membership checks). *By construction,* $V_k \subseteq \mathcal{P}^k(V_0)$ *and* $E_k \subseteq \mathcal{P}(V_k) \setminus \{\varnothing\}$ *for* $k = 1, 2, 3$. *Thus* $\mathrm{SHG}^{(3)} = (V_3, E_3)$ *satisfies Definition 5.*

Interpretation. *Level 0 lists columns; level 1 models relational schemas and key constraints; level 2 groups tables into logical databases; level 3 groups databases into services and specifies inter-service dependencies as superedges.*

2.2. Property Graph

A Property Graph is a directed multigraph in which both vertices and edges may carry arbitrary key–value attributes, and where edges are additionally assigned labels. This structure enables schema-flexible modeling of relational data [6–8, 36]. Below, we provide a concise formal definition together with illustrative examples.

Definition 6 (Property Graph)**.** *(cf.[6–8, 36]) Fix three (possibly infinite) sets*

$$\Sigma \quad \textit{(edge–label alphabet)},$$

$$K \quad \textit{(property keys)},$$
$$S \quad \textit{(property values)}.$$

A property graph† *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

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

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

Remark 1. *(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 6 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 5 (Property Graph for a Streaming-Platform Dataset)**.** *We illustrate Definition 6 with a concrete, small-scale model of an on-line movie-streaming service.*

Label alphabet, keys, and value domain.

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

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

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

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

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.

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

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 6. 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.*

3. Results and Revisits: Property HyperGraphs

Property HyperGraphs generalize hypergraphs by allowing vertices and hyperedges to carry key–value properties and labels, supporting flexible modeling.

Notation 1. *Fix three (possibly infinite) sets*

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

and let $\perp \notin S$ *be a distinguished symbol.*

Definition 7 (Property HyperGraph)**.** *A* property hypergraph *is a quadruple*

$$H \;=\; \bigl(V,\, E,\, \lambda,\, \mu\bigr)$$

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

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

We write

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

Example 6 (Property HyperGraph for Patient–Symptom Dataset)**.** *We illustrate Definition 7 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.

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

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

$$S = \{\textit{“Constitutional”},\ \textit{“Respiratory”},\ \textit{“Neurological”}\} \,\cup\, \{\textit{Strings}\} \,\cup\, \mathbb{N}.$$

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:*

$$\lambda(e_1) = \texttt{COVID19}, \quad \mu(e_1, \texttt{age}) = 45,$$
$$\mu(e_1, \texttt{gender}) = \text{“Male”}, \quad \mu(e_1, \texttt{severity}) = \text{“Moderate”},$$
$$\lambda(e_2) = \texttt{Influenza}, \quad \mu(e_2, \texttt{age}) = 30,$$
$$\mu(e_2, \texttt{gender}) = \text{“Female”}, \quad \mu(e_2, \texttt{severity}) = \text{“Mild”},$$
$$\lambda(e_3) = \texttt{Migraine}, \quad \mu(e_3, \texttt{age}) = 25,$$
$$\mu(e_3, \texttt{gender}) = \text{“Female”}, \quad \mu(e_3, \texttt{severity}) = \text{“Severe”}.$$

Keysets and values. *For example,*

$$(v_4) = \{\texttt{category}, \texttt{ICD\_code}\},$$
$$(v_4, \texttt{ICD\_code}) = \text{“R51”},$$
$$(e_2) = \{\texttt{age}, \texttt{gender}, \texttt{severity}\},$$
$$(e_2, \texttt{severity}) = \text{“Mild”}.$$

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies Definition 7, modelling a patient-symptom dataset in which each record links multiple symptoms and carries patient attributes.

Example 7 (Property HyperGraph for University Course Enrollment)**.** *We illustrate Definition 7 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:*

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

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,*

$$(v_2) = \{\texttt{name}, \texttt{role}\}, \quad (v_2, \texttt{name}) = \text{“Alice”},$$

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

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies all clauses of Definition 7: 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 8 (Property HyperGraph for Film Productions)**.** *We illustrate Definition 7 by modelling a film-production scenario.*

Label alphabet, keys, and value domain.

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

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

$$S = \{\text{“Actor”}, \text{“Director”}, \text{“Producer”}\} \cup \mathit{Strings} \cup \mathbb{N}.$$

Vertices. *Let*

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

with properties

$$\mu(v_1, \mathit{name}) = \text{“Alice Johnson”},$$
$$\mu(v_1, \mathit{roleType}) = \text{“Actor”},$$
$$\mu(v_1, \mathit{birthYear}) = 1985,$$
$$\mu(v_2, \mathit{name}) = \text{“Bob Lee”},$$
$$\mu(v_2, \mathit{roleType}) = \text{“Actor”},$$
$$\mu(v_2, \mathit{birthYear}) = 1978,$$
$$\mu(v_3, \mathit{name}) = \text{“Carol Smith”},$$
$$\mu(v_3, \mathit{roleType}) = \text{“Director”},$$
$$\mu(v_3, \mathit{nationality}) = \text{“USA”},$$
$$\mu(v_4, \mathit{name}) = \text{“David Kumar”},$$
$$\mu(v_4, \mathit{roleType}) = \text{“Producer”},$$
$$\mu(v_4, \mathit{nationality}) = \text{“UK”},$$
$$\mu(v_5, \mathit{name}) = \text{“Eva Zhang”},$$
$$\mu(v_5, \mathit{roleType}) = \text{“Actor”},$$
$$\mu(v_5, \mathit{birthYear}) = 1990.$$

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:*

$$\lambda(e_1) = \mathtt{TheGreatAdventure}, \quad \mu(e_1, \mathtt{releaseYear}) = 2024,$$
$$\mu(e_1, \mathtt{genre}) = \text{“Action”}, \quad \mu(e_1, \mathtt{boxOffice}) = 120000000,$$
$$\lambda(e_2) = \mathtt{MysteryNight}, \quad \mu(e_2, \mathtt{releaseYear}) = 2023,$$
$$\mu(e_2, \mathtt{genre}) = \text{“Mystery”}, \quad \mu(e_2, \mathtt{boxOffice}) = 85000000.$$

Keysets and values. *For example,*

$$(v_3) = \{\mathtt{name}, \mathtt{roleType}, \mathtt{nationality}\},$$
$$(v_3, \mathtt{roleType}) = \text{“Director”},$$
$$(e_1) = \{\mathtt{releaseYear}, \mathtt{genre}, \mathtt{boxOffice}\},$$
$$(e_1, \mathtt{genre}) = \text{“Action”}.$$

The quadruple $H = (V, E, \lambda, \mu)$ thus satisfies all requirements of Definition 7: 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 1 (Generalisation of Property Graphs and Hypergraphs)**.** *Let $H = (V, E, \lambda, \mu)$ be a property hypergraph over $(\Sigma, K, S, \bot)$. 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 $(V, E, s, t, \lambda, \mu, \bot)$ is precisely a Property Graph as in Definition 6.

(ii) If $\Sigma = \{\sigma_0\}$ is a singleton and $\mu(x, k) \equiv \bot$ for all (x, k), then $H = (V, E)$ collapses to an ordinary Hypergraph as in Definition 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 6 follow immediately.

For (ii), since Σ has only one element, λ carries no additional information; and because $\mu \equiv \bot$, no vertex or hyperedge carries a property. Thus $H = (V, E)$ satisfies exactly the conditions of Definition 3, concluding the proof.

Theorem 2 (Vertex-induced sub-Property HyperGraph)**.** *Let $H = (V, E, \lambda, \mu)$ be a property hypergraph and let $U \subseteq V$. Define*

$$V_U := U, \qquad E_U := \{\, e \in E \mid e \subseteq U \,\},$$

and restrict the maps by

$$\lambda_U := \lambda|_{E_U} \quad \text{and} \quad \mu_U := \mu|_{(U \cup E_U) \times K}.$$

Then $H_U := (V_U, E_U, \lambda_U, \mu_U)$ is a property hypergraph.

Proof. We check the items of Definition 7. (a) By construction $V_U \subseteq V$. (b) Each $e \in E_U$ is nonempty and $e \subseteq U = V_U$, hence $E_U \subseteq \mathcal{P}(V_U) \setminus \{\emptyset\}$. (c) λ_U maps E_U into Σ by restriction. (d) μ_U maps $(V_U \cup E_U) \times K$ into $S \cup \{\bot\}$ by restriction. All conditions hold.

Theorem 3 (Edge-induced sub-Property HyperGraph). *Let $H = (V, E, \lambda, \mu)$ be a property hypergraph and let $F \subseteq E$. Set*

$$V_F := \bigcup_{e \in F} e, \qquad E_F := F,$$

and restrict $\lambda_F := \lambda|_{E_F}$ and $\mu_F := \mu|_{(V_F \cup E_F) \times K}$. Then $H_F := (V_F, E_F, \lambda_F, \mu_F)$ is a property hypergraph.

Proof. (a) $V_F \subseteq V$ by definition. (b) Every $e \in E_F$ is nonempty and satisfies $e \subseteq V_F$, hence $E_F \subseteq \mathcal{P}(V_F) \setminus \{\emptyset\}$. (c)–(d) Follow immediately from restricting λ and μ to the new domains/codomains.

Theorem 4 (Uniformity and inheritance). *Suppose $H = (V, E, \lambda, \mu)$ is such that there exists $k \geq 1$ with $|e| = k$ for all $e \in E$. Then H is k-*uniform*. Moreover, every subfamily $F \subseteq E$ yields a k-uniform edge-induced sub-Property HyperGraph H_F.*

Proof. By hypothesis, for each $e \in E$ we have $e \subseteq V$ and $|e| = k$, which is precisely the definition of k-uniformity. If $F \subseteq E$, then every $e \in F$ still satisfies $|e| = k$, so Theorem 3 gives a k-uniform sub-Property HyperGraph.

Theorem 5 (Disjoint union). *Let $H_1 = (V_1, E_1, \lambda_1, \mu_1)$ and $H_2 = (V_2, E_2, \lambda_2, \mu_2)$ be property hypergraphs with disjoint vertex sets $V_1 \cap V_2 = \varnothing$. Define*

$$V := V_1 \cup V_2, \qquad E := E_1 \cup E_2,$$

and the piecewise maps

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

Then $H := (V, E, \lambda, \mu)$ is a property hypergraph.

Proof. (a) V is a set of vertices. (b) Each $e \in E_i$ is nonempty and $e \subseteq V_i \subseteq V$, hence $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$. (c)–(d) Disjointness of V_1, V_2 makes λ and μ well-defined with the same codomains as before.

Theorem 6 (Incidence representation as a Property Graph). *Let $H = (V, E, \lambda, \mu)$ be a property hypergraph. Form a (directed) Property Graph*

$$\mathcal{G} = (V_G, E_G, s, t, \lambda_G, \mu_G, \bot)$$

by taking

$$V_G := V \ \cup \ E \quad (\textit{disjoint union of tags “vertex” and “edge”}),$$

$$E_G := \{\, (e,v) \in E \times V \mid v \in e \,\}, \qquad s(e,v) = e, \quad t(e,v) = v,$$

and define

$$\lambda_G(e,v) := \lambda(e) \in \Sigma, \qquad \mu_G(x,k) := \begin{cases} \mu(x,k), & x \in V \cup E \ (= V_G), \\ \bot, & x \in E_G. \end{cases}$$

Then $\mathcal{G}$ is a Property Graph (in the sense of a directed, edge-labelled, attributed multigraph).

Proof. By construction V_G is a set and $E_G \subseteq V_G \times V_G$. The maps s, t are projections, hence well-defined. The edge-labelling λ_G uses the hyperedge label $\lambda(e)$; the property map μ_G agrees with μ on all vertices of $\mathcal{G}$ (i.e. original vertices and hyperedges) and assigns the sentinel $\bot$ to edges, ensuring $\mu_G : (V_G \cup E_G) \times K \to S \cup \{\bot\}$. Thus all clauses of the Property Graph definition are satisfied.

Theorem 7 (Forgetting labels and properties)**.** *If $\Sigma = \{\sigma_0\}$ is a singleton and $\mu \equiv \bot$, then any property hypergraph $H = (V, E, \lambda, \mu)$ reduces to the ordinary hypergraph (V, E).*

Proof. With Σ a singleton, λ carries no information beyond existence, and with $\mu \equiv \bot$ there are no attributes on vertices or hyperedges. The remaining data are precisely V and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$, which is the definition of a hypergraph.

4. Main Result: Property SuperHyperGraphs

We introduce and formalise the concept of a Property SuperHyperGraph. Property SuperHyperGraphs extend SuperHyperGraphs with labels and key–value attributes on supervertices and superedges, supporting hierarchical attributed modeling.

Definition 8 (Iterated powerset (Recall))**.** *Let V_0 be a finite, nonempty base set. Define*

$$\mathcal{P}^0(V_0) := V_0, \qquad \mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \quad (k \in \mathbb{N}).$$

For $n \in \mathbb{N}$, the elements of $\mathcal{P}^n(V_0)$ are called level-n carriers *(or n-*carriers*).*

Definition 9 (Property n-SuperHyperGraph)**.** *Fix alphabets/sets*

$$\Sigma \ (\textit{edge-label alphabet}), \qquad K \ (\textit{property keys}), \qquad S \ (\textit{property values}),$$

and a distinguished symbol $\bot \notin S$. For $n \in \mathbb{N}$, a property n-SuperHyperGraph (over V_0) *is a quadruple*

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

consisting of:

(i) a vertex set $V^{(n)} \subseteq \mathcal{P}^n(V_0)$*;*

(ii) a finite family of nonempty superedges $E^{(n)} \subseteq \mathcal{P}\big(V^{(n)}\big) \setminus \{\varnothing\}$*;*

(iii) an edge-labelling map $\lambda : E^{(n)} \to \Sigma$*;*

(iv) a property map

$$\mu : \Big(D^{(n)} \times K\Big) \longrightarrow S \cup \{\bot\}, \qquad D^{(n)} := \Big(\bigcup_{k=0}^{n} \mathcal{P}^k(V_0)\Big) \ \cup \ E^{(n)},$$

interpreted by $\mu(x,k) = s \in S$ *when object* x *carries key* k *with value* s*, and* $\mu(x,k) = \bot$ *otherwise.*

We set

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

Elements of $V^{(n)}$ *are* n-supervertices*; elements of* $E^{(n)}$ *are* n-superedges.

Example 9 (Property 3-SuperHyperGraph for an Image-Classification Dataset)**.** *We organise a hierarchical dataset with images (level 0), species classes (level 1), genera (level 2), and dataset splits (level 3).*

Level 0 (images). *Let*

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

Assign base-level properties (allowed since $V_0 \subseteq D^{(3)}$*):*

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

Level 1 (species; 1-supervertices).

$$C_{Cat} = \{i_1, i_2\}, \quad C_{Dog} = \{i_3, i_4\}, \quad C_{Bird} = \{i_5, i_6\} \ \in \ \mathcal{P}^1(V_0).$$

Set $V^{(1)} = \{C_{Cat}, C_{Dog}, C_{Bird}\}$ *and*

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

Level 2 (genera; 2-supervertices).

$$G_{Felidae} = \{C_{Cat}\}, \quad G_{Canidae} = \{C_{Dog}\}, \quad G_{Passeridae} = \{C_{Bird}\} \ \in \ \mathcal{P}^2(V_0),$$

with $V^{(2)} = \{G_{Felidae}, G_{Canidae}, G_{Passeridae}\}$ *and*

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

Level 3 (splits; 3-supervertices).

$$S_{Train} = \{G_{Felidae}, G_{Canidae}\}, \qquad S_{Test} = \{G_{Passeridae}\} \in \mathcal{P}^3(V_0),$$

with $V^{(3)} = \{S_{Train}, S_{Test}\}$ *and*

$$\mu(S_{Train}, \texttt{split}) = \text{“Train”},\ \mu(S_{Train}, \texttt{fraction}) = 0.8,$$

$$\mu(S_{Test}, \texttt{split}) = \text{“Test”},\ \mu(S_{Test}, \texttt{fraction}) = 0.2.$$

3-superedges. *Let*

$$e_{Full} = \{S_{Train}, S_{Test}\} \in \mathcal{P}(V^{(3)}) \setminus \{\varnothing\},$$

$$E^{(3)} = \{\, e_{Full} \,\}.$$

With $\Sigma = \{\texttt{ImageNetSubset}\}$,

$$\lambda(e_{Full}) = \texttt{ImageNetSubset}, \qquad \mu(e_{Full}, \texttt{version}) = \text{“v1.0”}.$$

Keysets. *For instance,* $(C_{Dog}) = \{\texttt{species}, \texttt{count}\}$ *and* $(C_{Dog}, \texttt{species}) =$ *“Canis lupus”.*
Thus $H^{(3)} = (V^{(3)}, E^{(3)}, \lambda, \mu)$ *satisfies Definition 9.*

Example 10 (Property 3-SuperHyperGraph for Multi-Tier Database Systems)**.** *We model columns (level 0), tables (1), schemas (2), and clusters (3).*

Level 0 (columns).

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

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

Level 1 (tables; 1-supervertices).

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

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

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

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

$$\mu(T_{\mathsf{Users}}, \texttt{tableName}) = \text{“Users”},\ \mu(T_{\mathsf{Users}}, \texttt{rowCount}) = 120000,$$

$$\mu(T_{\mathsf{Orders}}, \texttt{tableName}) = \text{“Orders”},\ \mu(T_{\mathsf{Orders}}, \texttt{rowCount}) = 450000,$$

$$\mu(T_{\mathsf{Items}}, \texttt{tableName}) = \text{“Items”},\ \mu(T_{\mathsf{Items}}, \texttt{rowCount}) = 950000.$$

Level 2 (schemas; 2-supervertices).

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

$\mu(S_{Sales}, \texttt{schemaName}) =$ *"SalesDB"*, $\mu(S_{Sales}, \texttt{version}) =$ *"v1.2"*.

Level 3 (clusters; 3-supervertices).

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

with

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

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

3-superedges. *Let* $e_{Enterprise} = \{C_{Primary}, C_{Replica}\}$, $E^{(3)} = \{e_{Enterprise}\}$, $\Sigma = \{\texttt{EnterpriseDB}\}$, *and set*

$$\lambda(e_{Enterprise}) = \texttt{EnterpriseDB},$$

$$\mu(e_{Enterprise}, \texttt{admin}) = \text{"DBA Team"},$$

$$\mu(e_{Enterprise}, \texttt{uptime\_SLA}) = 99.99.$$

Then $H^{(3)} = (V^{(3)}, E^{(3)}, \lambda, \mu)$ *is a property* 3*-SHG.*

Example 11 (A 3-level Property SuperHyperGraph for Urban Transit)**.** *We model an urban transit system with stops (level 0), routes (level 1), lines (level 2), and networks (level 3).*

Level 0 (base stops). *Let*

$$V_0 = \{\, s_1, s_2, s_3, s_4, s_5, s_6 \,\}.$$

Assign stop properties (allowed since $V_0 \subseteq D^{(3)}$*):*

$$\mu(s_1, \texttt{zone}) = \text{"1"},\ \mu(s_1, \texttt{lat}) = 35.69,\ \mu(s_1, \texttt{lon}) = 139.70,$$

$$\mu(s_2, \texttt{zone}) = \text{"1"},\ \ldots$$

Level 1 (routes as sets of stops; 1-supervertices). *Define*

$$R_A = \{s_1, s_2, s_3\}, \qquad R_B = \{s_3, s_4, s_5\},$$

$$R_C = \{s_5, s_6\} \ \in\ \mathcal{P}^1(V_0).$$

Set $V^{(1)} = \{R_A, R_B, R_C\}$ *and assign*

$$\mu(R_A, \texttt{routeID}) = \text{"A"},\ \mu(R_A, \texttt{headwayMin}) = 6,$$

$$\mu(R_B, \texttt{routeID}) = \text{"B"},\ \mu(R_B, \texttt{headwayMin}) = 8,$$

$$\mu(R_C, \texttt{routeID}) = \text{"C"},\ \mu(R_C, \texttt{headwayMin}) = 12.$$

Level 2 (lines as sets of routes; 2-supervertices). *Let*

$$L_{Red} = \{R_A, R_B\}, \qquad L_{Blue} = \{R_C\} \;\in\; \mathcal{P}^2(V_0).$$

Set $V^{(2)} = \{L_{Red}, L_{Blue}\}$ *with*

$$\mu(L_{Red}, \texttt{color}) = \text{"Red"},\ \mu(L_{Red}, \texttt{operator}) = \text{"MetroCo"},$$

$$\mu(L_{Blue}, \texttt{color}) = \text{"Blue"},\ \mu(L_{Blue}, \texttt{operator}) = \text{"MetroCo"}.$$

Level 3 (networks as sets of lines; 3-supervertices). *Define*

$$N_{City} = \{L_{Red}, L_{Blue}\},$$

$$N_{Regional} = \{L_{Blue}\} \;\in\; \mathcal{P}^3(V_0).$$

Set $V^{(3)} = \{N_{City}, N_{Regional}\}$ *and assign*

$$\mu(N_{City}, \texttt{name}) = \text{"City Network"},\ \mu(N_{City}, \texttt{rev}) = \text{"2025Q3"},$$

$$\mu(N_{Regional}, \texttt{name}) = \text{"Regional Link"}.$$

3-superedges and labels. *Let the label alphabet* $\Sigma = \{\texttt{FareAgreement}, \texttt{Interchange}\}$*. Define*

$$e_1 = \{N_{City}, N_{Regional}\}, \qquad e_2 = \{N_{City}\},$$

and

$$E^{(3)} = \{e_1, e_2\} \;\subseteq\; \mathcal{P}\big(V^{(3)}\big) \setminus \{\varnothing\}.$$

Set

$$\lambda(e_1) = \texttt{FareAgreement},\ \mu(e_1, \texttt{year}) = 2025,$$

$$\lambda(e_2) = \texttt{Interchange},\ \mu(e_2, \texttt{hub}) = \text{"Central"}.$$

Verification. *Each route is a subset of* $V_0 \Rightarrow V^{(1)} \subseteq \mathcal{P}^1(V_0)$*. Each line is a set of routes* $\Rightarrow V^{(2)} \subseteq \mathcal{P}^2(V_0)$*. Each network is a set of lines* $\Rightarrow V^{(3)} \subseteq \mathcal{P}^3(V_0)$*. Finally,* e_1, e_2 *are nonempty subsets of* $V^{(3)}$*. Thus*

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

is a property 3-SuperHyperGraph.

Example 12 (A 3-level Property SuperHyperGraph for Academic Collections)**.** *We model academic artifacts with papers (level 0), venues (level 1), fields (level 2), and programs/consortia (level 3).*

Level 0 (papers). *Let*

$$V_0 = \{\, p_1, p_2, p_3, p_4, p_5 \,\}.$$

Assign basic properties:

$$\mu(p_1, \texttt{title}) = \text{“A Study on X”},\ \mu(p_1, \texttt{year}) = 2023,$$

$$\mu(p_2, \texttt{title}) = \text{“Learning Y”},\ \mu(p_2, \texttt{year}) = 2024,\ \ldots$$

Level 1 (venues; 1-supervertices). *Group papers by publication venue:*

$$V_{\text{ConfA}} = \{p_1, p_2\}, \qquad V_{\text{ConfB}} = \{p_3, p_4\}, \qquad V_{\text{JournalC}} = \{p_5\} \ \in\ \mathcal{P}^1(V_0).$$

Set $V^{(1)} = \{V_{\text{ConfA}}, V_{\text{ConfB}}, V_{\text{JournalC}}\}$ *and assign*

$$\mu(V_{\text{ConfA}}, \texttt{abbr}) = \text{“ConfA”},\ \mu(V_{\text{ConfA}}, \texttt{ISSN}) = \text{“1111-1111”},$$

$$\mu(V_{\text{ConfB}}, \texttt{abbr}) = \text{“ConfB”},\ \mu(V_{\text{ConfB}}, \texttt{ISSN}) = \text{“2222-2222”},$$

$$\mu(V_{\text{JournalC}}, \texttt{abbr}) = \text{“JrnC”},\ \mu(V_{\text{JournalC}}, \texttt{ISSN}) = \text{“3333-3333”}.$$

Level 2 (fields; 2-supervertices). *Group venues by field:*

$$F_{\text{ML}} = \{V_{\text{ConfA}}, V_{\text{ConfB}}\},$$

$$F_{\text{Theory}} = \{V_{\text{JournalC}}\} \ \in\ \mathcal{P}^2(V_0),$$

with $V^{(2)} = \{F_{\text{ML}}, F_{\text{Theory}}\}$ *and*

$$\mu(F_{\text{ML}}, \texttt{fieldName}) = \text{“Machine Learning”},\ \mu(F_{\text{ML}}, \texttt{hIndex}) = 120,$$

$$\mu(F_{\text{Theory}}, \texttt{fieldName}) = \text{“Theory”},\ \mu(F_{\text{Theory}}, \texttt{hIndex}) = 85.$$

Level 3 (programs/consortia; 3-supervertices).

$$P_{\text{AIAlliance}} = \{F_{\text{ML}}\},$$

$$P_{\text{SciCouncil}} = \{F_{\text{Theory}}\} \ \in\ \mathcal{P}^3(V_0),$$

and $V^{(3)} = \{P_{\text{AIAlliance}}, P_{\text{SciCouncil}}\}$ *with*

$$\mu(P_{\text{AIAlliance}}, \texttt{program}) = \text{“AI Alliance”},$$

$$\mu(P_{\text{SciCouncil}}, \texttt{program}) = \text{“Science Council”}.$$

3-superedges and labels. *Let* $\Sigma = \{\texttt{Consortium}, \texttt{DataSharing}\}$*. Define*

$$e_{\text{cons}} = \{P_{\text{AIAlliance}}, P_{\text{SciCouncil}}\},$$

$$e_{\text{share}} = \{P_{\text{AIAlliance}}\},$$

$$E^{(3)} = \{e_{\text{cons}}, e_{\text{share}}\} \;\subseteq\; \mathcal{P}\big(V^{(3)}\big) \setminus \{\varnothing\}.$$

Set

$$\lambda(e_{\text{cons}}) = \texttt{Consortium},\;\; \mu(e_{\text{cons}}, \texttt{agreementYear}) = 2025,$$

$$\lambda(e_{\text{share}}) = \texttt{DataSharing},\;\; \mu(e_{\text{share}}, \texttt{scope}) = \text{“anonymized-metadata”}.$$

Verification. *Each venue is a subset of* $V_0 \Rightarrow V^{(1)} \subseteq \mathcal{P}^1(V_0)$*. Each field is a set of venues* $\Rightarrow V^{(2)} \subseteq \mathcal{P}^2(V_0)$*. Each program is a set of fields* $\Rightarrow V^{(3)} \subseteq \mathcal{P}^3(V_0)$*. The listed edges are nonempty subsets of* $V^{(3)}$*. Hence*

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

is a property 3*-SuperHyperGraph.*

Theorem 8 (Generalisation)**.** *Let* $H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$ *be a property* n*-SuperHyperGraph over the base* V_0*. Then:*

- *(i) If* $n = 1$*, every* $e \in E^{(1)}$ *has* $|e| = 2$*, and an orientation is chosen on each such pair, then* $H^{(1)}$ *induces a Property Graph in the sense of Definition 6.*
- *(ii) If* $n = 1$*,* $\Sigma = \{\sigma_0\}$ *is a singleton and* $\mu \equiv \perp$*, then* $H^{(1)}$ *reduces to the classical hypergraph* $\big(V^{(1)}, E^{(1)}\big)$*.*
- *(iii) If* $\Sigma = \{\sigma_0\}$ *and* $\mu \equiv \perp$ *(for arbitrary* n*), then* $H^{(n)}$ *reduces to the ordinary* n*-SuperHyperGraph* $\big(V^{(n)}, E^{(n)}\big)$*.*

Proof. (1) Since $E^{(1)} \subseteq \mathcal{P}(V^{(1)})$ and $|e| = 2$ for each $e \in E^{(1)}$, every edge is a 2–element subset $\{u, v\} \subseteq V^{(1)}$. Fix, once and for all, a choice of orientation for each unordered pair:

$$\theta:\; E^{(1)} \longrightarrow V^{(1)} \times V^{(1)}, \qquad \theta(\{u, v\}) = (u, v) \;\text{ or }\; (v, u).$$

Define the directed edge set

$$E_G := \{\; \theta(e) \mid e \in E^{(1)} \;\} \;\subseteq\; V^{(1)} \times V^{(1)}.$$

Set $V_G := V^{(1)}$ and define source/target maps by $s(u, v) = u$, $t(u, v) = v$. Let $\pi : E_G \to E^{(1)}$ be the forgetful map $\pi(u, v) = \{u, v\}$. Define the label and property maps by

$$\lambda_G(a) := \lambda\big(\pi(a)\big) \in \Sigma, \qquad \mu_G(x, k) := \begin{cases} \mu(x, k), & x \in V_G = V^{(1)}, \\ \mu\big(\pi(x), k\big), & x \in E_G. \end{cases}$$

Then $\mathcal{G} = (V_G, E_G, s, t, \lambda_G, \mu_G, \perp)$ satisfies all clauses of Definition 6: V_G is the vertex set, $E_G \subseteq V_G \times V_G$, s, t are well-defined, λ_G labels directed edges via their underlying undirected pairs, and μ_G is a property map with codomain $S \cup \{\perp\}$ (using the original μ on vertices and pulling back along π on edges). Hence $H^{(1)}$ yields a Property Graph.

(2) If $n = 1$, $\Sigma = \{\sigma_0\}$ and $\mu \equiv \perp$, then labels carry no information and no vertex/edge has a nontrivial property. The structure $H^{(1)}$ reduces to its underlying pair $(V^{(1)}, E^{(1)})$, with $E^{(1)} \subseteq \mathcal{P}(V^{(1)}) \setminus \{\varnothing\}$, which is precisely the definition of a hypergraph.

(3) The same argument applies verbatim for general n: with a trivial label alphabet and $\mu \equiv \perp$, only $V^{(n)} \subseteq \mathcal{P}^n(V_0)$ and $E^{(n)} \subseteq \mathcal{P}(V^{(n)}) \setminus \{\varnothing\}$ remain. This is exactly an n-SuperHyperGraph.

Theorem 9 (Induced sub-n-SuperHyperGraph)**.** *Let $H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$ be a property n-SuperHyperGraph over the base V_0, and let $U_0 \subseteq V_0$. Define*

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

Let

$$D_U^{(n)} := \Big(\bigcup_{r=0}^{n} \mathcal{P}^r(U_0)\Big) \;\cup\; E_U^{(n)},$$

and set $H_U^{(n)} := \big(V_U^{(n)}, E_U^{(n)}, \lambda|_{E_U^{(n)}}, \mu|_{D_U^{(n)} \times K}\big)$. Then $H_U^{(n)}$ is a property n-SuperHyperGraph over U_0.

Proof. We verify each clause of the definition.

(i) Vertex condition. We first note the *monotonicity of iterated powersets*: if $A \subseteq B$, then $\mathcal{P}^r(A) \subseteq \mathcal{P}^r(B)$ for all $r \geq 0$. This is proved by induction on r: for $r = 0$ it is $A \subseteq B$; if $\mathcal{P}^r(A) \subseteq \mathcal{P}^r(B)$, then $\mathcal{P}^{r+1}(A) = \mathcal{P}\big(\mathcal{P}^r(A)\big) \subseteq \mathcal{P}\big(\mathcal{P}^r(B)\big) = \mathcal{P}^{r+1}(B)$.

Applying this with $A = U_0 \subseteq B = V_0$ and $r = n-1$, we get $\mathcal{P}^{n-1}(U_0) \subseteq \mathcal{P}^{n-1}(V_0)$. Hence every $x \in V_U^{(n)}$ lies in $\mathcal{P}^n(U_0)$ (because x is by definition a subset of $\mathcal{P}^{n-1}(U_0)$), so $V_U^{(n)} \subseteq \mathcal{P}^n(U_0)$.

(ii) Edge condition. By construction, every $e \in E_U^{(n)}$ satisfies $e \subseteq V_U^{(n)}$ and $e \neq \varnothing$ (because $E^{(n)}$ consisted of nonempty subsets of $V^{(n)}$, and we only removed vertices). Thus $E_U^{(n)} \subseteq \mathcal{P}\big(V_U^{(n)}\big) \setminus \{\varnothing\}$.

(iii) Label map. The restriction $\lambda|_{E_U^{(n)}} : E_U^{(n)} \to \Sigma$ is well-defined with the same codomain Σ.

(iv) Property map. Since $D_U^{(n)} \subseteq \big(\bigcup_{r=0}^{n} \mathcal{P}^r(V_0)\big) \cup E^{(n)}$ by monotonicity and by the definition of $E_U^{(n)}$, the restricted map $\mu|_{D_U^{(n)} \times K} : D_U^{(n)} \times K \to S \cup \{\perp\}$ is well-defined with the same codomain.

All axioms are satisfied, hence $H_U^{(n)}$ is a property n-SuperHyperGraph over U_0.

Example 13 (Induced sub-n-SuperHyperGraph: a concrete $n = 2$ instance)**.** *Fix*

$$\Sigma = \{\texttt{Link}\}, \qquad K = \{\texttt{tag}\}, \qquad S = \mathbb{N} \cup \mathit{Strings}, \qquad \perp \notin S,$$

and let the base be $V_0 = \{a, b, c, d\}$. Define 1-level carriers (subsets of V_0)

$$A_1 = \{a, b\}, \quad A_2 = \{b, c\}, \quad A_3 = \{c, d\} \in \mathcal{P}^1(V_0),$$

and 2-supervertices

$$G_1 = \{A_1, A_2\}, \qquad G_2 = \{A_2, A_3\} \in \mathcal{P}^2(V_0).$$

Set $V^{(2)} = \{G_1, G_2\} \subseteq \mathcal{P}^2(V_0)$ and

$$E^{(2)} = \{\, e \,\}, \qquad e = \{G_1, G_2\} \in \mathcal{P}(V^{(2)}) \setminus \{\varnothing\}.$$

Label and properties:

$$\lambda(e) = \texttt{Link}, \qquad \mu(G_1, \texttt{tag}) = \text{"north"}, \quad \mu(G_2, \texttt{tag}) = \text{"south"}, \quad \mu(e, \texttt{tag}) = \text{"overlap"}.$$

Thus $H^{(2)} = (V^{(2)}, E^{(2)}, \lambda, \mu)$ is a property 2-SHG over V_0.

Now take $U_0 = \{a, b, c\} \subseteq V_0$. Then $\mathcal{P}^1(U_0) = \mathcal{P}(U_0)$, and we check membership:

$$A_1, A_2 \subseteq U_0 \quad \Rightarrow \quad G_1 \subseteq \mathcal{P}^1(U_0), \qquad A_3 \nsubseteq U_0 \quad \Rightarrow \quad G_2 \nsubseteq \mathcal{P}^1(U_0).$$

Hence

$$V_U^{(2)} = \{G_1\}, \qquad E_U^{(2)} = \varnothing,$$

and with $D_U^{(2)} = \mathcal{P}^0(U_0) \cup \mathcal{P}^1(U_0) \cup \mathcal{P}^2(U_0) \cup E_U^{(2)}$ we obtain the induced substructure

$$H_U^{(2)} = \big(V_U^{(2)}, E_U^{(2)}, \lambda|_{E_U^{(2)}}, \mu|_{D_U^{(2)} \times K}\big),$$

which is (by Theorem 9) a property 2-SuperHyperGraph over the base U_0.

Theorem 10 (Flattening projection). *Let*

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

be a property n-SuperHyperGraph with $n \geq 1$. Define the vertex-flattening map

$$\varphi_V : \ \mathcal{P}^n(V_0) \longrightarrow \mathcal{P}^{n-1}(V_0),$$

$$\varphi_V(X) = \bigcup_{x \in X} x,$$

and its action on edges by image

$$\varphi_E : \ \mathcal{P}\big(V^{(n)}\big) \setminus \{\varnothing\} \longrightarrow \mathcal{P}\big(\varphi_V[V^{(n)}]\big) \setminus \{\varnothing\},$$

$$\varphi_E(e) = \{\ \varphi_V(v) \mid v \in e\ \}.$$

Assume the following compatibility on fibres*:*

$$\text{(labels on edges)} \quad \varphi_E(e) = \varphi_E(e') \implies \lambda(e) = \lambda(e'),$$

$$\textit{(properties on vertices)}\quad \varphi_V(v)=\varphi_V(v') \implies \mu(v,k)=\mu(v',k)\ \ \forall k\in K,$$
$$\textit{(properties on edges)}\quad \varphi_E(e)=\varphi_E(e') \implies \mu(e,k)=\mu(e',k)\ \ \forall k\in K.$$

Then with

$$V' := \varphi_V[V^{(n)}] \subseteq \mathcal{P}^{n-1}(V_0),$$
$$E' := \{\ \varphi_E(e) \mid e\in E^{(n)}\ \} \subseteq \mathcal{P}(V')\setminus\{\varnothing\},$$

and with

$$\lambda'(\varphi_E(e)) := \lambda(e), \qquad \mu'(z',k) := \begin{cases}\mu(v,k), & z'=\varphi_V(v)\in V',\\ \mu(e,k), & z'=\varphi_E(e)\in E',\end{cases}$$

the quadruple $H'=(V',E',\lambda',\mu')$ is a property $(n-1)$-SuperHyperGraph over V_0.

Proof. (i) Vertex condition. By definition of φ_V, we have $\varphi_V(X)\in\mathcal{P}^{n-1}(V_0)$ for every $X\in\mathcal{P}^n(V_0)$. Hence $V'=\varphi_V[V^{(n)}]\subseteq\mathcal{P}^{n-1}(V_0)$.

(ii) Edge condition. Let $e\in E^{(n)}$. Since $e\subseteq V^{(n)}$ and $e\neq\varnothing$, its image $\varphi_E(e)=\{\varphi_V(v)\mid v\in e\}$ is a nonempty subset of V'. Thus $E'\subseteq\mathcal{P}(V')\setminus\{\varnothing\}$.

(iii) Labels. If $\varphi_E(e)=\varphi_E(e')$, the label-compatibility assumption gives $\lambda(e)=\lambda(e')$, so λ' is well-defined on E' and $\lambda':E'\to\Sigma$.

(iv) Properties. If $\varphi_V(v)=\varphi_V(v')$ then $\mu(v,k)=\mu(v',k)$ for all $k\in K$, and if $\varphi_E(e)=\varphi_E(e')$ then $\mu(e,k)=\mu(e',k)$ for all $k\in K$. Hence μ' is well-defined on $(V'\cup E')\times K$ with codomain $S\cup\{\perp\}$.

All clauses of the definition of a property $(n-1)$-SuperHyperGraph are satisfied.

Example 14 (Flattening projection: a concrete $2\to1$ instance)**.** *Let the base set be $V_0=\{a,b,c\}$. Consider the property 2-SuperHyperGraph*

$$H^{(2)}=(V^{(2)},E^{(2)},\lambda,\mu)$$

with

$$V^{(2)}=\Big\{v_1=\big\{\{a,b\},\{b,c\}\big\},\quad v_2=\big\{\{a\},\{a,c\}\big\}\Big\}\subseteq\mathcal{P}^2(V_0),$$
$$E^{(2)}=\big\{\ e=\{v_1,v_2\}\ \big\}\subseteq\mathcal{P}(V^{(2)})\setminus\{\varnothing\}.$$

Let the label alphabet be $\Sigma=\{\alpha\}$ and set $\lambda(e)=\alpha$. Fix keys $K=\{\texttt{tag},\texttt{weight}\}$ and a value domain S containing strings and integers; take $\perp\notin S$. Define the property map by

$$\mu(v_1,\texttt{tag})=\text{“T1”},\qquad \mu(v_2,\texttt{tag})=\text{“T2”},\qquad \mu(e,\texttt{weight})=3,$$

and $\mu(\cdot,\cdot)=\perp$ for all other key–object pairs.

The vertex–flattening map $\varphi_V:\mathcal{P}^2(V_0)\to\mathcal{P}^1(V_0)$ yields

$$\varphi_V(v_1)=\{a,b\}\cup\{b,c\}=\{a,b,c\},\qquad \varphi_V(v_2)=\{a\}\cup\{a,c\}=\{a,c\}.$$

The induced edge map φ_E gives

$$\varphi_E(e)=\big\{\varphi_V(v_1),\varphi_V(v_2)\big\}=\big\{\{a,b,c\},\{a,c\}\big\}.$$

Hence the flattened structure

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

is a property 1*-SuperHyperGraph with*

$$V' = \varphi_V[V^{(2)}] = \{\{a,b,c\},\{a,c\}\} \subseteq \mathcal{P}^1(V_0), \qquad E' = \{\, \varphi_E(e) \,\}.$$

By the theorem, we set

$$\lambda'(\varphi_E(e)) = \lambda(e) = \alpha, \quad \mu'(\{a,b,c\}, \texttt{tag}) = \mu(v_1, \texttt{tag}) = \text{“T1”}, \quad \mu'(\{a,c\}, \texttt{tag}) = \mu(v_2, \texttt{tag}) = \text{“T2”},$$

$$\mu'(\varphi_E(e), \texttt{weight}) = \mu(e, \texttt{weight}) = 3,$$

and $\mu'(\cdot,\cdot) = \bot$ *otherwise. Since no two distinct vertices (resp. edges) of* $H^{(2)}$ *share the same* φ_V*-image (resp.* φ_E*-image), the fibre–compatibility conditions hold trivially, and Theorem 10 applies.*

Theorem 11 (Iterated flattening)**.** *With notation as above, for* $1 \le k \le n$ *define*

$$\varphi_V^{(k)} := \underbrace{\varphi_V \circ \cdots \circ \varphi_V}_{k\ \textit{times}} : \mathcal{P}^n(V_0) \to \mathcal{P}^{n-k}(V_0),$$

and extend to edges by $\varphi_E^{(k)}(e) := \{\ \varphi_V^{(k)}(v) \mid v \in e\ \}$*. If the label/property compatibilities hold at each intermediate stage, then*

$$H^{(n-k)} := \big(\varphi_V^{(k)}[V^{(n)}],\ \{\varphi_E^{(k)}(e) \mid e \in E^{(n)}\},\ \lambda^{(k)},\ \mu^{(k)}\big)$$

is a property $(n-k)$*-SuperHyperGraph.*

Proof. Fix $n \in \mathbb{N}$ and $1 \le k \le n$. Recall the vertex–flattening map $\varphi_V : \mathcal{P}^r(V_0) \to \mathcal{P}^{r-1}(V_0),\ X \mapsto \bigcup_{x\in X} x$ for $r \ge 1$, and its extension to (nonempty) edge-sets $\varphi_E : \mathcal{P}(V^{(r)}) \setminus \{\varnothing\} \to \mathcal{P}(\varphi_V[V^{(r)}]) \setminus \{\varnothing\},\ \varphi_E(e) := \{\ \varphi_V(v) \mid v \in e\ \}$. For $k \ge 1$, define the k-fold compositions

$$\varphi_V^{(k)} := \underbrace{\varphi_V \circ \cdots \circ \varphi_V}_{k\ \text{times}} : \mathcal{P}^n(V_0) \longrightarrow \mathcal{P}^{n-k}(V_0), \qquad \varphi_E^{(k)}(e) := \{\ \varphi_V^{(k)}(v) \mid v \in e\ \}.$$

We prove by induction on k that

$$H^{(n-k)} := \Big(V^{(n-k)}, E^{(n-k)}, \lambda^{(k)}, \mu^{(k)}\Big) := \Big(\varphi_V^{(k)}[V^{(n)}],\ \{\varphi_E^{(k)}(e) \mid e \in E^{(n)}\},\ \lambda^{(k)},\ \mu^{(k)}\Big)$$

is a property $(n-k)$-SuperHyperGraph, provided the label/property compatibilities hold at each intermediate stage.

Base case $k = 1$*.* Set

$$V' := \varphi_V[V^{(n)}] \subseteq \mathcal{P}^{n-1}(V_0), \qquad E' := \{\ \varphi_E(e) \mid e \in E^{(n)}\ \} \subseteq \mathcal{P}(V') \setminus \{\varnothing\}.$$

The inclusion $V' \subseteq \mathcal{P}^{n-1}(V_0)$ is immediate from the codomain of φ_V. Nonemptiness of each $\varphi_E(e)$ holds because $e \neq \varnothing$ and φ_V is total.

By the assumed *edge-label compatibility*

$$\varphi_E(e) = \varphi_E(e') \implies \lambda(e) = \lambda(e'),$$

the map $\lambda' : E' \to \Sigma,\ \lambda'(\varphi_E(e)) := \lambda(e)$ is well-defined.

By the assumed *property compatibility* (for both vertices and edges)

$$\varphi_V(v) = \varphi_V(v') \implies \mu(v,k) = \mu(v',k), \qquad \varphi_E(e) = \varphi_E(e') \implies \mu(e,k) = \mu(e',k),$$

the map $\mu' : (V' \cup E') \times K \to S \cup \{\perp\}$ defined by

$$\mu'(\varphi_V(v),k) := \mu(v,k), \qquad \mu'(\varphi_E(e),k) := \mu(e,k)$$

is well-defined. Consequently (V', E', λ', μ') is a property $(n-1)$-SuperHyperGraph.

Induction step. Assume the claim holds for some k with $1 \leq k < n$: that is,

$$H^{(n-k)} = \Big(V^{(n-k)}, E^{(n-k)}, \lambda^{(k)}, \mu^{(k)}\Big) = \Big(\varphi_V^{(k)}[V^{(n)}],\ \{\varphi_E^{(k)}(e) \mid e \in E^{(n)}\},\ \lambda^{(k)},\ \mu^{(k)}\Big)$$

is a property $(n-k)$-SuperHyperGraph.

By the hypothesis "label/property compatibilities hold at each intermediate stage", we have, for the $(k+1)$-st stage, the compatibilities on $H^{(n-k)}$:

$$\varphi_V(x) = \varphi_V(y) \implies \mu^{(k)}(x,\cdot) = \mu^{(k)}(y,\cdot),$$

$$\varphi_E(a) = \varphi_E(b) \implies \lambda^{(k)}(a) = \lambda^{(k)}(b),\quad \mu^{(k)}(a,\cdot) = \mu^{(k)}(b,\cdot).$$

Applying the base case to $H^{(n-k)}$ yields a property $(n-k-1)$-SuperHyperGraph

$$\widehat{H} \;=\; \Big(\varphi_V[V^{(n-k)}],\ \{\varphi_E(e^\star) \mid e^\star \in E^{(n-k)}\},\ \widehat{\lambda},\ \widehat{\mu}\Big).$$

Using $V^{(n-k)} = \varphi_V^{(k)}[V^{(n)}]$ and $E^{(n-k)} = \{\varphi_E^{(k)}(e) \mid e \in E^{(n)}\}$, we compute

$$\varphi_V[V^{(n-k)}] \;=\; \varphi_V\big[\varphi_V^{(k)}[V^{(n)}]\big] \;=\; \varphi_V^{(k+1)}[V^{(n)}],$$

and for each $e \in E^{(n)}$,

$$\varphi_E\big(\varphi_E^{(k)}(e)\big) \;=\; \big\{\ \varphi_V(w)\ \big|\ w \in \{\varphi_V^{(k)}(v) \mid v \in e\}\ \big\} \;=\; \big\{\ \varphi_V^{(k+1)}(v)\ \big|\ v \in e\ \big\} \;=\; \varphi_E^{(k+1)}(e).$$

Hence $\widehat{H}$ coincides with

$$\Big(\varphi_V^{(k+1)}[V^{(n)}],\ \{\varphi_E^{(k+1)}(e) \mid e \in E^{(n)}\},\ \lambda^{(k+1)},\ \mu^{(k+1)}\Big),$$

where $\lambda^{(k+1)}$ and $\mu^{(k+1)}$ are the induced maps defined by

$$\lambda^{(k+1)}\big(\varphi_E^{(k+1)}(e)\big) := \lambda(e),$$

$$\mu^{(k+1)}\big(\varphi_V^{(k+1)}(v), k_0\big) := \mu(v, k_0),$$
$$\mu^{(k+1)}\big(\varphi_E^{(k+1)}(e), k_0\big) := \mu(e, k_0),$$

which are well-defined by the stated compatibilities at each stage. Thus the statement holds for $k+1$.

By induction on k, the structure

$$H^{(n-k)} = \Big(\varphi_V^{(k)}[V^{(n)}],\ \{\varphi_E^{(k)}(e) \mid e \in E^{(n)}\},\ \lambda^{(k)},\ \mu^{(k)}\Big)$$

is a property $(n-k)$-SuperHyperGraph for all $1 \le k \le n$.

Example 15 (Iterated flattening: a concrete $3 \to 1$ instance with $k=2$)**.** *Let the base set be $V_0 = \{x, y\}$. Build a property 3-SuperHyperGraph*

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

as follows. Level–1 carriers are $\{x\}, \{y\}, \{x, y\}$. Define two level–2 carriers

$$U_1 = \big\{\{x\}, \{y\}\big\}, \qquad U_2 = \big\{\{x, y\}\big\},$$

and two level–3 supervertices

$$w_1 = \{U_1\}, \qquad w_2 = \{U_2\}.$$

Set

$$V^{(3)} = \{w_1, w_2\} \subseteq \mathcal{P}^3(V_0), \qquad E^{(3)} = \big\{\ e = \{w_1, w_2\}\ \big\}.$$

Take $\Sigma = \{\sigma\}$ with $\lambda(e) = \sigma$, and let the property map be trivial: $\mu(\cdot, \cdot) \equiv \bot$. (This choice satisfies the fibre–compatibility conditions automatically.)

First flattening ($k = 1$). *The map $\varphi_V : \mathcal{P}^3(V_0) \to \mathcal{P}^2(V_0)$ gives*

$$\varphi_V(w_1) = U_1 = \big\{\{x\}, \{y\}\big\}, \qquad \varphi_V(w_2) = U_2 = \big\{\{x, y\}\big\}.$$

Thus

$$V_{(1)}^{(2)} = \varphi_V[V^{(3)}] = \{U_1, U_2\}, \qquad E_{(1)}^{(2)} = \big\{\ \varphi_E(e) = \{U_1, U_2\}\ \big\}.$$

With $\lambda^{(1)}\big(\{U_1, U_2\}\big) = \sigma$ and $\mu^{(1)} \equiv \bot$, Theorem 10 yields a property 2-SuperHyperGraph $H_{(1)}^{(2)}$.

Second flattening ($k = 2$). *Apply φ_V again, now $\varphi_V : \mathcal{P}^2(V_0) \to \mathcal{P}^1(V_0)$:*

$$\varphi_V(U_1) = \{x\} \cup \{y\} = \{x, y\}, \qquad \varphi_V(U_2) = \{x, y\}.$$

Hence both level–2 vertices collapse to the same level–1 carrier, and

$$V_{(2)}^{(1)} = \big\{\{x, y\}\big\}, \qquad E_{(2)}^{(1)} = \big\{\ \varphi_E^{(2)}(e) = \{\{x, y\}\}\ \big\}.$$

Define $\lambda^{(2)}(\{\{x,y\}\}) = \sigma$ *and keep* $\mu^{(2)} \equiv \perp$. *The fibre–compatibility assumptions are satisfied (both labels constant and properties trivial), so by Theorem 11 with* $k = 2$ *we obtain a property* 1*-SuperHyperGraph*

$$H^{(1)} = \left(V^{(1)}_{(2)}, E^{(1)}_{(2)}, \lambda^{(2)}, \mu^{(2)}\right) = \left(\{\{x,y\}\}, \{\{\{x,y\}\}\}, \sigma, \perp\right).$$

Theorem 12 (Uniformity)**.** *Let* $H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$ *be a property* n*-SuperHyperGraph. Assume there exists an integer* $k \geq 1$ *such that* $|e| = k$ *for every* $e \in E^{(n)}$*. Then* $H^{(n)}$ *is* k*-uniform, i.e., every level-*n *superedge links exactly* k n*-supervertices. Moreover, any subfamily* $E'^{(n)} \subseteq E^{(n)}$ *inherits* k*-uniformity.*

Proof. By hypothesis, for each $e \in E^{(n)}$ we have $e \subseteq V^{(n)}$ and $|e| = k$. By the usual definition, this is precisely the statement that $H^{(n)}$ is k-uniform. Now let $E'^{(n)} \subseteq E^{(n)}$. For any $e \in E'^{(n)}$ we still have $e \subseteq V^{(n)}$ and (because $e \in E^{(n)}$) $|e| = k$. Hence the restriction $(V^{(n)}, E'^{(n)}, \lambda|_{E'^{(n)}}, \mu|_{(V^{(n)} \cup E'^{(n)}) \times K})$ is again k-uniform. No further conditions are required.

Theorem 13 (Disjoint union)**.** *Let* $H^{(n)}_i = (V^{(n)}_i, E^{(n)}_i, \lambda_i, \mu_i)$ *be property* n*-SuperHyperGraphs over pairwise disjoint base sets* $V^{(i)}_0$ *for* $i = 1, 2$ *(so* $V^{(1)}_0 \cap V^{(2)}_0 = \varnothing$*). To avoid accidental identifications (e.g. of* $\varnothing$*) across levels, form tagged copies*

$$\iota^{(r)}_i : \mathcal{P}^r(V^{(i)}_0) \longrightarrow \mathcal{P}^r\big(V^{(1)}_0 \cup V^{(2)}_0\big) \times \{i\}, \quad x \mapsto (x, i) \qquad (0 \leq r \leq n),$$

and define

$$V^{(n)} = \iota^{(n)}_1[V^{(n)}_1] \cup \iota^{(n)}_2[V^{(n)}_2], \qquad E^{(n)} = \{\, \iota^{(n)}_1[e] \mid e \in E^{(n)}_1 \,\} \cup \{\, \iota^{(n)}_2[e] \mid e \in E^{(n)}_2 \,\}.$$

Set the label and property maps by

$$\lambda\big(\iota^{(n)}_i[e]\big) := \lambda_i(e), \qquad \mu\big((\iota^{(r)}_i(x), i), k\big) := \mu_i(x, k),$$

for $e \in E^{(n)}_i$*,* $x \in \mathcal{P}^r(V^{(i)}_0)$*,* $0 \leq r \leq n$*. Then* $\big(V^{(n)}, E^{(n)}, \lambda, \mu\big)$ *is a property* n*-SuperHyperGraph over the disjoint base* $V^{(1)}_0 \cup V^{(2)}_0$ *(identifying* (v, i) *with* v *since the bases are disjoint).*

Proof. We verify the clauses of the definition.

(Vertices) Since $V^{(n)}_i \subseteq \mathcal{P}^n(V^{(i)}_0)$ for each i, we have $\iota^{(n)}_i[V^{(n)}_i] \subseteq \mathcal{P}^n(V^{(1)}_0 \cup V^{(2)}_0) \times \{i\}$. Hence $V^{(n)} \subseteq \mathcal{P}^n(V^{(1)}_0 \cup V^{(2)}_0) \times \{1, 2\}$, which (via the obvious identification) sits inside $\mathcal{P}^n(V^{(1)}_0 \cup V^{(2)}_0)$.

(Edges) For each $e \in E^{(n)}_i$ we know $e \subseteq V^{(n)}_i$ and $e \neq \varnothing$. Therefore $\iota^{(n)}_i[e] \subseteq \iota^{(n)}_i[V^{(n)}_i] \subseteq V^{(n)}$, and $\iota^{(n)}_i[e] \neq \varnothing$. Thus every element of $E^{(n)}$ is a nonempty subset of $V^{(n)}$.

(Labels) If $\iota^{(n)}_i[e] = \iota^{(n)}_j[e']$, then necessarily $i = j$ and $e = e'$ (by tagging), so the piecewise definition $\lambda(\iota^{(n)}_i[e]) = \lambda_i(e)$ is well-defined and maps into Σ.

(Properties) For any $x \in \bigcup_{r=0}^{n} \mathcal{P}^r(V_0^{(i)})$ or $x \in E_i^{(n)}$, the tag ensures that $(\iota_i^{(r)}(x), i)$ determines the source structure uniquely, so $\mu\big((\iota_i^{(r)}(x), i), k\big) := \mu_i(x, k)$ is well-defined with codomain $S \cup \{\perp\}$.

All axioms are therefore satisfied, so $(V^{(n)}, E^{(n)}, \lambda, \mu)$ is a property n-SuperHyperGraph. If one prefers literal unions without tags, the same construction applies because the base sets are disjoint; tagging is used only to preclude pathological coincidences such as $\varnothing$.

Example 16 (Disjoint union: a concrete $n = 1$ instance)**.** *Fix common alphabets*

$$\Sigma = \{\alpha, \beta\}, \qquad K = \{\texttt{info}\}, \qquad S = \textit{Strings}, \qquad \perp \notin S.$$

First component $H_1^{(1)}$. *Base $V_0^{(1)} = \{x_1, x_2\}$ (disjoint from the other base). Take*

$$V_1^{(1)} = \big\{A = \{x_1\},\ B = \{x_1, x_2\}\big\} \subseteq \mathcal{P}^1\big(V_0^{(1)}\big),$$

$$E_1^{(1)} = \{e_1\}, \qquad e_1 = \{A, B\}, \quad \lambda_1(e_1) = \alpha,$$

$$\mu_1(A, \texttt{info}) = \text{“single”}, \ \ \mu_1(B, \texttt{info}) = \text{“pair”}, \ \ \mu_1(e_1, \texttt{info}) = \text{“includes”}.$$

Second component $H_2^{(1)}$. *Base $V_0^{(2)} = \{y_1, y_2, y_3\}$ with $V_0^{(1)} \cap V_0^{(2)} = \varnothing$. Let*

$$V_2^{(1)} = \big\{C = \{y_1, y_2\},\ D = \{y_2, y_3\}\big\} \subseteq \mathcal{P}^1\big(V_0^{(2)}\big),$$

$$E_2^{(1)} = \{e_2\}, \qquad e_2 = \{C, D\}, \quad \lambda_2(e_2) = \beta,$$

$$\mu_2(C, \texttt{info}) = \text{“left”}, \ \ \mu_2(D, \texttt{info}) = \text{“right”}, \ \ \mu_2(e_2, \texttt{info}) = \text{“adjacent”}.$$

Tagged disjoint union. *Using the injections $\iota_i^{(1)} : \mathcal{P}^1(V_0^{(i)}) \to \mathcal{P}^1(V_0^{(1)} \cup V_0^{(2)}) \times \{i\}$, form*

$$V^{(1)} = \iota_1^{(1)}[V_1^{(1)}] \ \cup \ \iota_2^{(1)}[V_2^{(1)}], \qquad E^{(1)} = \{\iota_1^{(1)}[e_1],\ \iota_2^{(1)}[e_2]\}.$$

Define

$$\lambda\big(\iota_1^{(1)}[e_1]\big) = \alpha, \quad \lambda\big(\iota_2^{(1)}[e_2]\big) = \beta,$$

and for $x \in V_j^{(1)} \cup E_j^{(1)}$,

$$\mu\big((\iota_j^{(1)}(x), j), \texttt{info}\big) := \mu_j(x, \texttt{info}) \qquad (j = 1, 2).$$

Then

$$H^{(1)} = \big(V^{(1)}, E^{(1)}, \lambda, \mu\big)$$

is a property 1-SuperHyperGraph (Property HyperGraph) over the disjoint base $V_0^{(1)} \cup V_0^{(2)}$ by Theorem 13. If one suppresses tags, disjointness of the bases still prevents accidental identifications.

Theorem 14 (Line graph as a Property Graph)**.** *Let $H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$ be a property n-SuperHyperGraph. Define its line graph $\mathcal{G}$ by*

$$V_G := E^{(n)}, \qquad E_G := \{ (e_1, e_2) \in V_G \times V_G \mid e_1 \neq e_2,\ e_1 \cap e_2 \neq \varnothing \}.$$

Let $s(e_1, e_2) = e_1$ and $t(e_1, e_2) = e_2$, set $\lambda_G(e_1, e_2) := \lambda(e_1) \in \Sigma$, and define a property map

$$\mu_G(x, k) := \begin{cases} \mu(x, k), & x \in V_G (= E^{(n)}), \\ \perp, & x \in E_G. \end{cases}$$

Then $\mathcal{G} = (V_G, E_G, s, t, \lambda_G, \mu_G, \perp)$ is a Property Graph.

Proof. We check each item of the Property Graph definition.

(1) *Vertices and edges.* By construction $V_G = E^{(n)}$. Every element of E_G is an ordered pair of distinct superedges that intersect, so $E_G \subseteq V_G \times V_G$.

(2) *Source/target maps.* The maps $s, t : E_G \to V_G$ defined by $s(e_1, e_2) = e_1$ and $t(e_1, e_2) = e_2$ are well-defined by the previous item.

(3) *Edge labels.* For $(e_1, e_2) \in E_G$ we set $\lambda_G(e_1, e_2) = \lambda(e_1) \in \Sigma$. This is a bona fide edge-labelling (variants such as $(\lambda(e_1), \lambda(e_2)) \in \Sigma \times \Sigma$ are also admissible).

(4) *Properties.* On vertices $V_G = E^{(n)}$ we reuse μ; on edges we assign the sentinel $\perp$. Thus $\mu_G : (V_G \cup E_G) \times K \to S \cup \{\perp\}$ has the correct codomain, and $\perp \notin S$ by assumption.

All requirements are met; hence $\mathcal{G}$ is a Property Graph.

5. Conclusion and Outlook

This study has demonstrated how the expressive power of Property Graphs can be lifted to the richer settings of HyperGraphs and SuperHyperGraphs. We provided rigorous, uniform definitions for these generalizations and presented initial results that underscore their modeling potential. For clarity and ease of reference, Table 2 summarizes the concepts introduced and defined in this paper.

Future work may explore further extensions based on advanced uncertainty formalisms—including Fuzzy Sets[37], Vague Sets[38], Intuitionistic Fuzzy Sets[39], Paraconsistent Set[40], Soft Sets[41, 42], Picture Fuzzy Set[43, 44], Rough Sets[45, 46], Neutrosophic Sets (and their quadri-partitioned variants) [47, 48], Hesitant Fuzzy Sets [49], and Plithogenic Sets [50, 51]. 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. Also an exciting future direction would be to extend Property SuperHyperGraphs over complex algebraic structures such as Gaussian integers, Eisenstein integers, Quaternion integers, and even Octonion algebras (cf.[52, 53]). These extensions could yield novel frameworks for secure data modeling, multi-dimensional coding theory, and advanced cryptographic schemes, exploiting the non-commutative, associative, or non-associative properties of such algebras.

Table 2: Concise overview of Property Graph, Property Hypergraph, and Property n-SuperHyperGraph.

Model	Labels	Properties	Notes
Property Graph	$\lambda : E \to \Sigma$	$\mu : (V \cup E) \times K \to S \cup \{\perp\}$	Directed multigraph $G = (V, E)$; vertices/edges carry attributes; edges are labelled.
Property Hypergraph	$\lambda : E \to \Sigma$	$\mu : (V \cup E) \times K \to S \cup \{\perp\}$	Hypergraph $H = (V, E)$; hyperedges connect arbitrary vertex subsets; attributes and labels allowed.
Property n-SuperHyperGraph	$\lambda : E^{(n)} \to \Sigma$	$\mu : (D^{(n)} \times K) \to S \cup \{\perp\}$	$H^{(n)} = (V^{(n)}, E^{(n)}, \lambda, \mu)$; supervertices at level n, edges between them; supports hierarchical, attributed structures.

Funding

This work was carried out without any external financial support.

Acknowledgements

We are grateful to all colleagues and reviewers whose insights and encouragement have strengthened this work. We also thank the authors of the cited literature for laying the groundwork that enabled our study, and acknowledge the institutions that provided the resources and infrastructure essential for its completion.

Author's Contributions

Conceptualization, Takaaki Fujita; Investigation, Takaaki Fujita; Methodology, Takaaki Fujita; Writing – original draft, Takaaki Fujita; Writing – review & editing, All authors.

Data Availability

This is a theoretical study and did not produce any data. We invite future researchers to perform empirical validations of the concepts presented herein.

Ethical Approval

No human participants or animal subjects were involved in this research; ethical approval was therefore not required.

Conflicts of Interest

The authors declare that there are no conflicts of interest related to this study.

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

The ideas and models presented in this paper are theoretical and have not yet been empirically tested. While we have made every effort to ensure accuracy and proper citation, unintentional errors may remain. Readers should verify sources independently. The views expressed are those of the authors and do not necessarily reflect those of their institutions.

Consent to Publish

All authors have reviewed and approved this manuscript for submission.

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] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173(33):12, 2005.

[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] 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.

[12] Yifan Feng, Jiashu Han, Shihui Ying, and Yue Gao. Hypergraph isomorphism computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] 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.

[14] 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.

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

[16] G Deepa, B Praba, and VM Chandrasekaran. A study on energy of an intuitionistic fuzzy directed graph. *Research Journal of Pharmacy and Technology*, 9(2):190–195, 2016.

[17] Yanna J Kraakman and Clara Stegehuis. Configuration models for random directed hypergraphs. *arXiv preprint arXiv:2402.06466*, 2024.

[18] Muhammad Akram and Anam Luqman. A new decision-making method based on bipolar neutrosophic directed hypergraphs. *Journal of Applied Mathematics and Computing*, 57:547–575, 2018.

[19] Muhammad Akram and Anam Luqman. Certain networks models using single-valued neutrosophic directed hypergraphs. *Journal of Intelligent & Fuzzy Systems*, 33(1):575–588, 2017.

[20] Takaaki Fujita. Review of some superhypergraph classes: Directed, bidirected, soft, and rough. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*, 2024.

[21] Graeme Simsion and Graham Witt. *Data modeling essentials*. Elsevier, 2004.

[22] Michael Blaha. *Patterns of data modeling*, volume 1. CRC press, 2010.

[23] André Ribeiro, Afonso Silva, Alberto Rodrigues da Silva, et al. Data modeling and data analytics: a survey from a big data perspective. *Journal of Software Engineering and Applications*, 8(12):617, 2015.

[24] 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.

[25] Bibin K Jose and Zsolt Tuza. Hypergraph domination and strong independence.

Applicable Analysis and Discrete Mathematics, 3(2):347–358, 2009.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] Yasar Nacaroglu, Nihat Akgunes, Sedat Pak, and I Naci Cangul. Some graph parameters of power set graphs. *Advances & Applications in Discrete Mathematics*, 26(2), 2021.

[32] MA Shalu and S Devi Yamini. Counting maximal independent sets in power set graphs. *Indian Institute of Information Technology Design & Manufacturing (IIITD&M) Kancheepuram, India*, 2014.

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

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

[35] 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.

[36] 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.

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

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

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

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

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

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

[43] 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.

[44] 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.

[45] 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.

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

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

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

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

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

[51] 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.

[52] Muhammad Sajjad, Tariq Shah, Maha Alammari, and Huda Alsaud. Construction and decoding of bch-codes over the gaussian field. *IEEE Access*, 11:71972–71980, 2023.

[53] Muhammad Sajjad, Tariq Shah, Qin Xin, and Bander Almutairi. Eisenstein field bch codes construction and decoding. *AIMS Math*, 8(12):29453–29473, 2023.