Neutrosophic Knowledge, Vol. 6, 2025

University of New Mexico

Directed Acyclic SuperHypergraphs (DASH): A General Framework for Hierarchical Dependency Modeling

Takaaki Fujita[1] *

[1] Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan. Takaaki.fujita060@gmail.com

*Correspondence: Takaaki.fujita060@gmail.com

Received: *03 01, 2025;* Accepted: *05 20, 2025*

Abstract. Graph theory models pairwise relationships through vertices and edges, while hypergraphs generalize this framework by allowing hyperedges to connect multiple vertices simultaneously. SuperHyperGraphs, introduced by Smarandache [26, 27], extend hypergraphs via iterated powerset constructions. Directed Acyclic Graphs (DAGs) are cycle-free directed graphs widely used for dependency modeling, and Directed Acyclic Hypergraphs (DAHs) further generalize DAGs by capturing multi-way dependencies [18, 23]. In this paper, we introduce *Directed Acyclic SuperHypergraphs (DASH)*, which unify and extend both DAGs and DAHs within the SuperHyperGraph framework. We present a formal definition of DASH, characterize acyclicity through directed superhyperedges, and establish fundamental properties such as the existence of source supervertices and topological orderings. Our work provides a rigorous theoretical foundation for hierarchical dependency modeling in complex systems and paves the way for future advances in both the theory and applications of Hypergraph and SuperHyperGraph Theory.

Keywords: Superhypergraph, Hypergraph, Directed graph, Directed Hypergraph, Directed Superhypergraph, Directed Acyclic graph, Directed Acyclic Hypergraph

1. Preliminaries and Definitions

This section provides an introduction to the foundational concepts and definitions required for the discussions in this paper. For fundamental operations, concepts, and principles of graphs, refer to [7, 17]. Throughout this paper, we assume that all graphs are finite.

1.1. *Graph and Hypergraph*

Graph theory is the study of mathematical structures consisting of vertices connected by edges, used to model various types of relationships. A hypergraph is a generalized graph concepts that extends traditional graph concepts by allowing hyperedges, which connect multiple vertices rather than just pairs, enabling more complex relationships between elements [8, 9]. The basic definitions of graphs and hypergraphs are provided below.

Definition 1.1 (Graph). [6] A graph G is a mathematical structure consisting of a set of vertices $V(G)$ and a set of edges $E(G)$ that connect pairs of vertices, representing relationships or connections between them. Formally, a graph is defined as $G = (V, E)$, where V is the vertex set and E is the edge set.

Definition 1.2 (Cycle in a Graph). [6] A cycle in a graph $G = (V, E)$ is a path that starts and ends at the same vertex, with no repeated edges or vertices, except for the starting/ending vertex. Formally, a cycle is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that:

$$v_1 = v_k \quad \text{and} \quad (v_i, v_{i+1}) \in E \quad \text{for all} \quad 1 \leq i < k.$$

The length of the cycle is $k - 1$, which is the number of edges in the cycle.

Definition 1.3 (Hypergraph [2, 3]). A *hypergraph* $H = (V(H), E(H))$ is a pair where:

- $V(H)$: A non-empty set of vertices.
- $E(H)$: A set of hyperedges, each of which is a subset of $V(H)$.

This paper focuses exclusively on finite hypergraphs.

1.2. *SuperHyperGraph*

A *SuperHyperGraph* is an extension of the traditional concept of a hypergraph, recently introduced and actively investigated in the literature [4, 5, 11–13, 27]. It can be regarded as a graph-theoretical construct that integrates recursive structures into hypergraphs. A SuperHyperGraph is characterized by an iteratively generated structure known as the n-th powerset, obtained through repeated application of the powerset operation. The formal definition is presented below. Here, the parameter n is assumed to be a natural number.

Definition 1.4 (n-th Powerset). (cf. [25, 28])

The n-th powerset of a set H, denoted $P_n(H)$, is constructed iteratively. Beginning with the standard powerset, the process is defined as:

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

In a similar manner, the n-th non-empty powerset, represented as $P_n^*(H)$, is recursively defined as:

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

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

Definition 1.5 (n-SuperHyperGraph)**.** [26, 27] Let V_0 be a finite *base set* of vertices. For each $k \geq 0$, define the iterative powerset $\mathcal{P}^k(V_0)$ by

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

where $\mathcal{P}(\cdot)$ denotes the power set. An *n-SuperHyperGraph* is a pair

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

with

$$V \;\subseteq\; \mathcal{P}^n(V_0) \quad \text{and} \quad E \;\subseteq\; \mathcal{P}^n(V_0).$$

Each element of V is an *n-supervertex*, and each element of E is an *n-superedge*.

Example 1.6 (2-SuperHyperGraph: Corporate Collaboration Network)**.** Let the base set of employees be

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

Form the first iterated powerset $\mathcal{P}^1(V_0)$ to obtain teams:

$$\mathrm{Team}_1 = \{\mathrm{Alice}, \mathrm{Bob}\}, \quad \mathrm{Team}_2 = \{\mathrm{Carol}, \mathrm{Dave}\}, \quad \mathrm{Team}_3 = \{\mathrm{Eve}\}.$$

The second iterated powerset $\mathcal{P}^2(V_0) = \mathcal{P}(\{\mathrm{Team}_1, \mathrm{Team}_2, \mathrm{Team}_3\})$ consists of all subsets of these teams. We select two of them as our 2-supervertices:

$$V = \bigl\{\{\mathrm{Team}_1, \mathrm{Team}_2\},\ \{\mathrm{Team}_2, \mathrm{Team}_3\}\bigr\}.$$

Each 2-supervertex represents a division that groups two teams. To model a cross-division project involving both divisions, define the set of 2-superedges:

$$E = \Bigl\{\{\{\mathrm{Team}_1, \mathrm{Team}_2\},\ \{\mathrm{Team}_2, \mathrm{Team}_3\}\}\Bigr\}.$$

Then

$$\mathrm{SHT}^{(2)} = (V, E)$$

is a 2-SuperHyperGraph capturing the structure of corporate collaboration across divisions.

Definition 1.7 (n-SuperHypertree)**.** (cf. [16]) An *n-SuperHypertree (n-SHT)* is an n-SuperHyperGraph $\mathrm{SHT}_n = (V, E)$ that satisfies the following properties:

(1) *Host Tree Condition*: There exists a tree $T = (V_T, E_T)$, called the *host tree*, such that:
- The vertex set of T is $V_T = V$, where $V \subseteq \mathcal{P}^n(V_0)$.

- Each n-superedge $e \in E$ corresponds to a connected subtree $T_e \subseteq T$. Specifically, for each $e \in E$, there exists a subtree T_e such that:

$$\bigcup_{t \in V(T_e)} B_t \supseteq e,$$

where $B_t \subseteq V$ are subsets associated with the nodes of T.

(2) *Acyclicity Condition*: The host tree T must be acyclic, ensuring that SHT_n inherits the acyclic structure of T.

(3) *Connectedness Condition*: For any two n-supervertices $v, w \in V$, there must exist a sequence of n-superedges $e_1, e_2, \ldots, e_k \in E$ such that:

(a) $v \in e_1$ and $w \in e_k$.

(b) $e_i \cap e_{i+1} \neq \emptyset$ for all $1 \leq i < k$.

Example 1.8 (1-SuperHypertree: Regional Supply Chain Network)**.** Let the base set of locations be

$$V_0 = \{\text{Farm, Mill, Distributor, Retailer}\}.$$

For $n = 1$, we select the primary distribution hubs as our 1-supervertices:

$$V = \big\{\{\text{Farm}, \text{Mill}\},\ \{\text{Mill}, \text{Distributor}\},\ \{\text{Distributor}, \text{Retailer}\}\big\}.$$

We construct the host tree $T = (V_T, E_T)$ by taking $V_T = V$ and

$$E_T = \big\{(\{\text{Farm}, \text{Mill}\},\ \{\text{Mill}, \text{Distributor}\}),\ (\{\text{Mill}, \text{Distributor}\},\ \{\text{Distributor}, \text{Retailer}\})\big\}.$$

Next, we define the set of 1-superedges E, each corresponding to a connected subtree of T:

$$E = \Big\{\{\{\text{Farm}, \text{Mill}\},\ \{\text{Mill}, \text{Distributor}\}\},\ \{\{\text{Mill}, \text{Distributor}\},\ \{\text{Distributor}, \text{Retailer}\}\}\Big\}.$$

Here:

- The first superedge links the Farm–Mill hub to the Mill–Distributor hub.
- The second superedge links the Mill–Distributor hub to the Distributor–Retailer hub.

Since T is acyclic and each superedge induces a connected subtree, $\mathrm{SHT}_1 = (V, E)$ satisfies the host tree, acyclicity, and connectedness conditions. This 1-SuperHypertree thus models the hierarchical flow of goods through the regional supply chain.

1.3. *Directed hypergraph*

A directed graph consists of vertices connected by edges with assigned directions, indicating relationships. A directed hypergraph is a hypergraph generalization of a directed graph. Similar to undirected hypergraphs, directed hypergraphs have been extensively studied for their various derivatives and applications (cf. [19, 21, 22]). Its definition is provided below.

Definition 1.9 (Directed Graph)**.** [31] A *directed graph* (digraph) $G = (V, E)$ consists of:

- V: A finite set of vertices.
- $E \subseteq V \times V$: A set of directed edges, where each edge is an ordered pair (u, v) with $u, v \in V$.

The edge (u, v) indicates a directed connection from vertex u (source) to vertex v (target).

Definition 1.10 (Directed Hypergraph)**.** [1,14] A *Directed Hypergraph* H is a pair $H = (V, E)$, where:

- V is a finite set of vertices (or nodes).
- E is a finite set of hyperarcs. Each hyperarc $e \in E$ is an ordered pair $e = (\mathrm{Tail}(e), \mathrm{Head}(e))$, where:
  - $\mathrm{Tail}(e) \subseteq V$ is a non-empty subset of vertices, called the *tail* of the hyperarc.
  - $\mathrm{Head}(e) \in V$ is a single vertex, called the *head* of the hyperarc.

Properties.

- A hyperarc $e = (\mathrm{Tail}(e), \mathrm{Head}(e))$ connects all vertices in $\mathrm{Tail}(e)$ to the vertex $\mathrm{Head}(e)$.
- When $|\mathrm{Tail}(e)| = 1$ for all $e \in E$, the directed hypergraph reduces to a standard directed graph.

Definition 1.11 (Directed n-SuperHypergraph)**.** [10] Let V_0 be a finite base set, and define its k-th iterated power set

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

An *Directed n-SuperHypergraph* is an ordered pair

$$\mathrm{DSH}_n = (V, E),$$

where:

(1) $V \subseteq \mathcal{P}^n(V_0)$ is the set of *n-supervertices.* Each $v \in V$ may be
  - a single element in V_0,
  - a subset of V_0,
  - a nested subset up to depth n,
  - the empty set, or
  - a fuzzy/indeterminate set (cf. [30]), depending on the application.

(2) $E \subseteq \big(\mathcal{P}(V) \times \mathcal{P}(V)\big)$ is a set of *directed n-superhyperedges.* Each directed n-superhyperedge $e \in E$ is an ordered pair

$$e = \big(T(e), H(e)\big),$$

with

$$T(e) \subseteq V \quad \text{and} \quad H(e) \subseteq V,$$

where:

- $T(e)$ is called the *tail set*, representing the "source" supervertices.
- $H(e)$ is called the *head set*, representing the "target" supervertices.

Example 1.12 (Real-World Example: Directed 2-SuperHypergraph in an IoT Data Network)**.** Consider an IoT deployment consisting of four sensors:

$$V_0 = \{\mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3, \mathrm{S}_4\}.$$

The first iterated powerset $\mathcal{P}^1(V_0) = P(V_0)$ yields all possible sensor clusters, for example $\{\{S_1, S_2\}, \{S_3\}, \{S_2, S_4\}, \ldots\}$. The second iterated powerset $\mathcal{P}^2(V_0) = P\big(P(V_0)\big)$ consists of collections of these clusters. We select two such collections as our set of 2-supervertices:

$$V = \Big\{\{\{S_1, S_2\}, \{S_3, S_4\}\},\ \{\{S_1, S_3\}, \{S_2, S_4\}\}\Big\}.$$

Here each 2-supervertex represents a regional aggregation unit grouping two sensor clusters.

Next, we define the set of directed 2-superhyperedges E as follows:

$$e_1 = \Big(\{\{S_1, S_2\}, \{S_3, S_4\}\},\ \{\{S_1, S_3\}, \{S_2, S_4\}\}\Big),$$
$$e_2 = \Big(\{\{S_1, S_3\}, \{S_2, S_4\}\},\ \{\{S_1, S_2\}, \{S_3, S_4\}\}\Big).$$

In this model:

- e_1 represents a data-flow from the first regional aggregator $\{\{S_1, S_2\}, \{S_3, S_4\}\}$ to the second $\{\{S_1, S_3\}, \{S_2, S_4\}\}$.
- e_2 captures the reverse synchronization flow.

Thus, $\mathrm{DSH}_2 = (V, E)$ is a Directed 2-SuperHypergraph modeling high-level, multi-cluster data exchanges in an IoT system.

1.4. *Directed Acyclic Graph (DAG)*

A Directed Acyclic Graph (DAG) is a directed graph that contains no cycles, where edges represent dependencies. DAGs are commonly used in scheduling, data flow analysis, and optimization [20, 24]. DAGs play a crucial role in various applications, including dependency resolution, causal inference, and parallel computing. A related concept is the bidirected acyclic graph, which is also well studied [15].

Definition 1.13 (Directed Acyclic Graph (DAG))**.** [24,29] A *Directed Acyclic Graph (DAG)* is a directed graph $G = (V, E)$ that contains no directed cycles. That is, there does not exist a sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that:

$$(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k), (v_k, v_1) \in E.$$

Example 1.14 (Directed Acyclic Graph (DAG): Project Workflow)**.** Consider a software project composed of five tasks ordered by dependencies:

$$V = \{\text{Planning, Design, Implementation, Testing, Deployment}\}.$$

We model the precedence constraints by the edge set

$$E = \{(\text{Planning, Design}), (\text{Design, Implementation}),$$

$$(\text{Implementation, Testing}), (\text{Planning, Testing}), (\text{Testing, Deployment})\}.$$

Here:

- Planning $\to$ Design and Planning $\to$ Testing ensure that both design and test planning begin only after requirements are set.
- Design $\to$ Implementation enforces that coding starts after design is complete.
- Implementation $\to$ Testing and Testing $\to$ Deployment reflect the usual build–test–release cycle.

Since no sequence of edges leads back to an earlier task, this graph contains no directed cycles and thus is a valid DAG.

The concept of a Directed Acyclic Hypergraph (DAH), as a hypergraph-based generalization of a DAG, has also been studied in the literature [18, 23]. Its formal definition is presented below.

Definition 1.15 (Directed Acyclic Hypergraph (DAH))**.** (cf. [18, 23]) A **Directed Acyclic Hypergraph (DAH)** is a directed hypergraph $H = (V, E)$ that does not contain any **hypercycle**, which is defined as a sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that for each $i = 1, \ldots, k$, there exists a hyperedge $e_i = (e_{T_i}, e_{H_i}) \in E$ with:

$$v_i \in e_{T_i} \quad \text{and} \quad v_{i+1} \in e_{H_i}, \quad \text{where } v_{k+1} = v_1.$$

Example 1.16 (Directed Acyclic Hypergraph: Software Development Workflow)**.** Consider a software development pipeline modeled as a directed acyclic hypergraph $H = (V, E)$, where

$$V = \{\text{Req, Design, Impl, Plan, Test, Deploy}\}$$

represents the key activities:

- Req: Requirements gathering
- Design: System design
- Impl: Implementation (coding)
- Plan: Test plan preparation
- Test: Testing
- Deploy: Deployment

Define the set of hyperedges

$$E = \{e_1, e_2, e_3, e_4, e_5\},$$

where each $e_i = (T(e_i), H(e_i))$ models dependencies:

$$\begin{aligned}
e_1 &= \bigl(\{\text{Req}\},\ \{\text{Design}\}\bigr),\\
e_2 &= \bigl(\{\text{Design}\},\ \{\text{Impl}\}\bigr),\\
e_3 &= \bigl(\{\text{Req}\},\ \{\text{Plan}\}\bigr),\\
e_4 &= \bigl(\{\text{Impl, Plan}\},\ \{\text{Test}\}\bigr),\\
e_5 &= \bigl(\{\text{Test}\},\ \{\text{Deploy}\}\bigr).
\end{aligned}$$

Here:

- e_1 and e_3 represent that both design and test-plan creation depend on the requirement phase.
- e_2 indicates design must complete before implementation.
- e_4 is a hyperedge with two tails $(\text{Impl}, \text{Plan})$ leading to testing.
- e_5 models deployment after successful testing.

Since there is no sequence of hyperedges leading back to any starting activity, H contains no hypercycles and is therefore a Directed Acyclic Hypergraph.

2. **Results**

This section presents the results obtained in this study.

2.1. *Directed Acyclic SuperHypergraph*

A Directed Acyclic SuperHypergraph (DASH) is a hierarchical hypergraph with directed superhyperedges, containing no cycles, used for complex dependency modeling and multi-level data structures.

Definition 2.1 (Directed Cycle in a SuperHypergraph)**.** Let $\mathrm{DSH}_n = (V, E)$ be a Directed n-SuperHypergraph. A *directed cycle* is a sequence of distinct n-supervertices

$$v_1, v_2, \ldots, v_k \quad (k \geq 2)$$

for which there exist directed n-superhyperedges

$$e_1, e_2, \ldots, e_k \in E$$

satisfying

$$v_i \in T(e_i) \quad \text{and} \quad v_{i+1} \in H(e_i), \quad \text{for } i = 1, \ldots, k,$$

with the convention $v_{k+1} = v_1$.

Definition 2.2 (Directed Acyclic SuperHypergraph (DASH))**.** A Directed n-SuperHypergraph $\mathrm{DSH}_n = (V, E)$ is called a *Directed Acyclic SuperHypergraph (DASH)* if it contains no directed cycle (as in Definition 2.1).

Definition 2.3 (Incoming SuperHyperedge and Source)**.** For a supervertex $v \in V$ in a Directed n-SuperHypergraph $\mathrm{DSH}_n = (V, E)$, define its set of *incoming superhyperedges* by

$$\mathrm{In}(v) := \{e \in E \mid v \in H(e)\}.$$

A supervertex v is called a *source* if

$$\mathrm{In}(v) = \emptyset.$$

Example 2.4 (Directed Acyclic SuperHypergraph (DASH): Social Media Group Merging)**.** Let the base set of user groups be

$$V_0 = \{\mathrm{G}_A,\ \mathrm{G}_B,\ \mathrm{G}_C\}.$$

We take $n = 1$, so our 1-supervertices are all nonempty subsets of V_0:

$$V = \bigl\{\{\mathrm{G}_A\},\ \{\mathrm{G}_B\},\ \{\mathrm{G}_C\},\ \{\mathrm{G}_A, \mathrm{G}_B\},\ \{\mathrm{G}_B, \mathrm{G}_C\},\ \{\mathrm{G}_A, \mathrm{G}_C\},\ \{\mathrm{G}_A, \mathrm{G}_B, \mathrm{G}_C\}\bigr\}.$$

Define the set of directed superhyperedges E to model successive merge operations:

$$\begin{aligned}
e_1 &= \bigl(\{\{\mathrm{G}_A\}, \{\mathrm{G}_B\}\},\ \{\{\mathrm{G}_A, \mathrm{G}_B\}\}\bigr),\\
e_2 &= \bigl(\{\{\mathrm{G}_B\}, \{\mathrm{G}_C\}\},\ \{\{\mathrm{G}_B, \mathrm{G}_C\}\}\bigr),\\
e_3 &= \bigl(\{\{\mathrm{G}_A, \mathrm{G}_B\}, \{\mathrm{G}_C\}\},\ \{\{\mathrm{G}_A, \mathrm{G}_B, \mathrm{G}_C\}\}\bigr),\\
e_4 &= \bigl(\{\{\mathrm{G}_A\}, \{\mathrm{G}_C\}\},\ \{\{\mathrm{G}_A, \mathrm{G}_C\}\}\bigr).
\end{aligned}$$

Here:

- e_1 merges groups G_A and G_B.
- e_2 merges groups G_B and G_C.
- e_3 merges the combined group $\{G_A, G_B\}$ with G_C.
- e_4 merges G_A and G_C.

No sequence of these superhyperedges forms a cycle, so $\mathrm{DSH}_1 = (V, E)$ is a Directed Acyclic SuperHypergraph.

Example 2.5 (Directed Acyclic SuperHypergraph (DASH): Manufacturing Assembly Process)**.** Let the base set of components be

$$V_0 = \{\mathrm{A},\ \mathrm{B},\ \mathrm{C}\},$$

where $\mathrm{A}, \mathrm{B}, \mathrm{C}$ denote basic parts. Taking $n = 1$, our 1-supervertices are the nonempty subsets of V_0:

$$V = \bigl\{\{\mathrm{A}\},\ \{\mathrm{B}\},\ \{\mathrm{C}\},\ \{\mathrm{A}, \mathrm{B}\},\ \{\mathrm{B}, \mathrm{C}\},\ \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}\bigr\}.$$

Define directed superhyperedges E to represent assembly steps:

$$e_1 = \big(\{\{\mathrm{A}\},\{\mathrm{B}\}\},\ \{\{\mathrm{A},\mathrm{B}\}\}\big),$$
$$e_2 = \big(\{\{\mathrm{B}\},\{\mathrm{C}\}\},\ \{\{\mathrm{B},\mathrm{C}\}\}\big),$$
$$e_3 = \big(\{\{\mathrm{A},\mathrm{B}\},\{\mathrm{C}\}\},\ \{\{\mathrm{A},\mathrm{B},\mathrm{C}\}\}\big).$$

Here:

- e_1 assembles parts A and B into subassembly AB.
- e_2 assembles parts B and C into subassembly BC.
- e_3 combines subassembly AB with part C to produce the final product ABC.

No directed sequence of these superhyperedges returns to a previous supervertex, so $\mathrm{DSH}_1 = (V, E)$ is a Directed Acyclic SuperHypergraph modeling the assembly process.

Theorem 2.6 (Existence of a Source)**.** *In any finite Directed Acyclic SuperHypergraph* $\mathrm{DSH}_n = (V, E)$*, there exists at least one supervertex* $v \in V$ *with* $\mathrm{In}(v) = \emptyset$*.*

Proof. Assume, for the sake of contradiction, that every supervertex $v \in V$ has at least one incoming superhyperedge; that is, $\mathrm{In}(v) \neq \emptyset$ for all $v \in V$. Pick an arbitrary supervertex $v_1 \in V$. By assumption, there exists an edge $e_1 \in E$ such that $v_1 \in H(e_1)$. Choose a supervertex $v_2 \in T(e_1)$. Again, since v_2 has an incoming edge, there exists an edge $e_2 \in E$ with $v_2 \in H(e_2)$; choose $v_3 \in T(e_2)$. Continue in this manner. Since V is finite, by the pigeonhole principle, some vertex must eventually repeat. Let $v_i = v_j$ for some $i < j$. Then the sequence

$$v_i, v_{i+1}, \ldots, v_j = v_i$$

forms a directed cycle, contradicting the acyclicity of DSH_n. Hence, there must exist at least one supervertex v with no incoming superhyperedges. □

Theorem 2.7 (Topological Ordering)**.** *Every finite Directed Acyclic SuperHypergraph* $\mathrm{DSH}_n = (V, E)$ *admits a topological ordering of its supervertices.*

Proof. We prove this by induction on the number of supervertices $|V|$.

Base Case: If $|V| = 1$, the unique ordering is trivial.

Inductive Step: Assume that every Directed Acyclic SuperHypergraph with fewer than $|V|$ supervertices has a topological ordering. By Theorem 2.6, there exists a source $v \in V$ (i.e., $\mathrm{In}(v) = \emptyset$). Remove v from DSH_n along with all superhyperedges incident to v (that is, all $e \in E$ with $v \in T(e)$ or $v \in H(e)$); denote the resulting Directed Acyclic SuperHypergraph by $\mathrm{DSH}'_n = (V \setminus \{v\}, E')$. By the induction hypothesis, DSH'_n has a topological ordering. Prepending v to this ordering yields a topological ordering for the entire DSH_n. □

Theorem 2.8 (Partial Order Induced by Reachability)**.** *Let* $\mathrm{DSH}_n = (V, E)$ *be a Directed Acyclic SuperHypergraph. Define a binary relation* $\prec$ *on* V *by declaring*

$$u \prec v \quad \textit{if there exists a directed path from } u \textit{ to } v.$$

Then, $\prec$ *is a strict partial order on* V*.*

Proof. We show that the relation $\prec$ is irreflexive, transitive, and asymmetric.

(i) **Irreflexivity:** Suppose, toward a contradiction, that $v \prec v$ for some $v \in V$. Then there exists a directed path from v back to itself, which forms a directed cycle. This contradicts the acyclicity of DSH_n. Thus, $v \not\prec v$ for any $v \in V$.

(ii) **Transitivity:** If $u \prec v$ and $v \prec w$, then there exist a directed path from u to v and one from v to w. By concatenating these paths, we obtain a directed path from u to w, so $u \prec w$.

(iii) **Asymmetry:** Assume that $u \prec v$. If it were also the case that $v \prec u$, then by transitivity we would have $u \prec u$, contradicting irreflexivity. Hence, if $u \prec v$ then it cannot be that $v \prec u$.

Thus, $\prec$ is a strict partial order on V. □

Theorem 2.9 (Generalization Property of Directed Acyclic SuperHypergraphs)**.** *Let* $\mathrm{DASH}_n = (V, E)$ *be a Directed Acyclic SuperHypergraph with base set* V_0 *and* $n \geq 0$*. Then:*

(1) Every Directed Acyclic Hypergraph is a special case of a Directed Acyclic SuperHypergraph. In particular, if $H = (V, E)$ *is a Directed Acyclic Hypergraph, then by taking* $V_0 = V$ *(so that* $\mathcal{P}^0(V_0) = V$*) and interpreting each hyperedge* $e \in E$ *as a superhyperedge with tail* $T(e)$ *and head* $H(e)$*, we obtain a Directed Acyclic SuperHypergraph isomorphic to* H*.*

(2) Every Directed Acyclic Graph is a special case of a Directed Acyclic SuperHypergraph. Specifically, if $G = (V, E)$ *is a Directed Acyclic Graph where each edge* e *is an ordered pair* (u, v)*, then by defining for each edge the corresponding superhyperedge with*

$$T(e) = \{u\} \quad \textit{and} \quad H(e) = \{v\},$$

the resulting structure is a Directed Acyclic SuperHypergraph isomorphic to G*.*

Proof. We prove the two statements separately.

(1) Directed Acyclic Hypergraph ⇒ Directed Acyclic SuperHypergraph:
Let $H = (V, E)$ be a Directed Acyclic Hypergraph. By definition, V is a finite set and each hyperedge $e \in E$ is an ordered pair

$$e = (e_T, e_H)$$

with $e_T, e_H \subseteq V$. Note that setting $n = 0$ yields

$$\mathcal{P}^0(V_0) = V_0.$$

Choosing $V_0 = V$ guarantees that $V \subseteq \mathcal{P}^0(V_0)$. Now, interpret each hyperedge $e \in E$ as a superhyperedge in a Directed Acyclic SuperHypergraph $\text{DASH}_0 = (V, E)$ by keeping the same tail and head sets:

$$T(e) = e_T \quad \text{and} \quad H(e) = e_H.$$

Since the acyclicity property is inherent in H, the resulting structure is acyclic and hence a Directed Acyclic SuperHypergraph that is isomorphic to H.

(2) Directed Acyclic Graph $\Rightarrow$ Directed Acyclic SuperHypergraph:
Let $G = (V, E)$ be a Directed Acyclic Graph. Each edge $e \in E$ is an ordered pair (u, v) with $u, v \in V$. Again, taking $n = 0$ and $V_0 = V$ gives us $V \subseteq \mathcal{P}^0(V_0)$. For each edge $e = (u, v)$ in G, define a corresponding superhyperedge in the Directed Acyclic SuperHypergraph $\text{DASH}_0 = (V, E')$ by setting:

$$T(e) = \{u\} \quad \text{and} \quad H(e) = \{v\}.$$

The acyclic nature of G ensures that no directed cycle is introduced in the superhypergraph representation. Therefore, G is isomorphic to a Directed Acyclic SuperHypergraph. □

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.

Ethical Approval

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

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.

Authors' Contributions

The sole author designed, analysed, interpreted, and prepared the manuscript.

Research Integrity

The authors hereby confirm that, to the best of their knowledge, this manuscript is their original work, has not been published in any other journal, and is not currently under consideration for publication elsewhere at this stage.

Disclaimer (Note on Computational Tools)

No computer-assisted proof, symbolic computation, or automated theorem proving tools (e.g., Mathematica, SageMath, Coq, etc.) were used in the development or verification of the results presented in this paper. All proofs and derivations were carried out manually and analytically by the authors.

Disclaimer (Limitations and Claims)

The theoretical concepts presented in this paper have not yet been subject to practical implementation or empirical validation. Future researchers are invited to explore these ideas in applied or experimental settings. Although every effort has been made to ensure the accuracy of the content and the proper citation of sources, unintentional errors or omissions may persist. Readers should independently verify any referenced materials.

To the best of the authors' knowledge, all mathematical statements and proofs contained herein are correct and have been thoroughly vetted. Should you identify any potential errors or ambiguities, please feel free to contact the authors for clarification.

The results presented are valid only under the specific assumptions and conditions detailed in the manuscript. Extending these findings to broader mathematical structures may require additional research. The opinions and conclusions expressed in this work are those of the authors alone and do not necessarily reflect the official positions of their affiliated institutions.

Competing Interests

Author has declared that no competing interests exist.

Consent to Publish declaration

The author approved to Publish declarations.

References

[1] Giorgio Ausiello and Luigi Laura. Directed hypergraphs: Introduction and fundamental algorithms?a survey. *Theoretical Computer Science*, 658:293–306, 2017.

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

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

[4] Yan Cao. Integrating treesoft and hypersoft paradigms into urban elderly care evaluation: A comprehensive n-superhypergraph approach. *Neutrosophic Sets and Systems*, 85:852–873, 2025.

[5] Y. V. M. Cepeda, M. A. R. Guevara, E. J. J. Mogro, and R. P. Tizano. Impact of irrigation water technification on seven directories of the san juan-patoa river using plithogenic n-superhypergraphs based on environmental indicators in the canton of pujili, 2021. *Neutrosophic Sets and Systems*, 74:46–56, 2024.

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

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

[8] Song Feng, Emily Heath, Brett Jefferson, Cliff Joslyn, Henry Kvinge, Hugh D Mitchell, Brenda Praggastis, Amie J Eisfeld, Amy C Sims, Larissa B Thackray, et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1):287, 2021.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Takaaki Fujita. Unifying grain boundary networks and crystal graphs: A hypergraph and superhypergraph perspective in material sciences. *Asian Journal of Advanced Research and Reports*, 19(5):344–379, 2025.

[14] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201, 1993.

[15] Vaithianathan Geetha and Niladhuri Sreenath. High concurrency for continuously evolving oodbms. In *Distributed Computing and Internet Technology: 8th International Conference, ICDCIT 2012, Bhubaneswar, India, February 2-4, 2012. Proceedings 8*, pages 94–105. Springer, 2012.

[16] 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.

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

[18] Mohammad Ali Javidian, Zhiyu Wang, Linyuan Lu, and Marco Valtorta. On a hypergraph probabilistic graphical model. *Annals of Mathematics and Artificial Intelligence*, 88:1003–1033, 2020.

[19] Spencer Krieger and John D. Kececioglu. Shortest hyperpaths in directed hypergraphs for reaction pathway inference. *Journal of computational biology : a journal of computational molecular cell biology*, 2023.

[20] Ari M Lipsky and Sander Greenland. Causal directed acyclic graphs. *JAMA*, 327(11):1083–1084, 2022.

[21] Heechan Moon, Hyunju Kim, Sunwoo Kim, and Kijung Shin. Four-set hypergraphlets for characterization of directed hypergraphs. *ArXiv*, abs/2311.14289, 2023.

[22] Kazusato Oko, Shinsaku Sakaue, and Shin ichi Tanigawa. Nearly tight spectral sparsification of directed hypergraphs. In *International Colloquium on Automata, Languages and Programming*, 2023.

[23] Merten Popp, Sebastian Schlag, Christian Schulz, and Daniel Seemaier. Multilevel acyclic hypergraph partitioning. In *2021 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 1–15. SIAM, 2021.

[24] Ian Shrier and Robert W Platt. Reducing bias through directed acyclic graphs. *BMC medical research methodology*, 8:1–15, 2008.

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

[26] Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.

[27] 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.

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

[29] Thomas C Williams, Cathrine C Bach, Niels B Matthiesen, Tine B Henriksen, and Luigi Gagliardi. Directed acyclic graphs: a tool for causal studies in paediatrics. *Pediatric research*, 84(4):487–493, 2018.

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

[31] Ping Zhang and Gary Chartrand. Introduction to graph theory. *Tata McGraw-Hill*, 2:2–1, 2006.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the publisher and/or the editor(s). This publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.