A Note on Line and Total Directed SuperHyperGraphs, Line Bidirected Graphs, Line MultiDirected Graphs, and Related Structures

Takaaki Fujita [1] *

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

Abstract

Hypergraphs extend classical graphs by allowing *hyperedges* to connect any nonempty subset of vertices, thereby capturing complex group-level relationships. Superhypergraphs advance this framework by introducing recursively nested powerset layers, enabling the representation of hierarchical and self-referential links among hyperedges. A *line graph* encodes the adjacencies between edges of an original graph by transforming each edge into a vertex and connecting two vertices if their corresponding edges share a common endpoint. A *total graph* incorporates both the vertices and edges of the original graph as its own vertices, with edges representing adjacency or incidence between these entities.

Various extensions of these graph concepts exist that incorporate directional information, such as *Directed Graphs*, *Bidirected Graphs*, and *Multidirected Graphs*. In this paper, we investigate the notions of line graphs and total graphs within the settings of *Directed HyperGraphs*, *Directed SuperHyperGraphs*, *Bidirected Graphs*, and *Multidirected Graphs*.

Keywords: SuperHyperGraph, HyperGraph, Line Graph, Total Graph, Iterated line graph, Iterated total graph

Contents in this paper

The remainder of this paper is organized as follows.

1 Preliminaries

In this section, we review the key concepts and notation used throughout this paper. All graphs and HyperGraphs considered here are finite.

1.1 Hypergraphs and Superhypergraphs

Classical graph models often struggle to capture certain real-world structures, particularly those with hierarchical or multi-layered connections. To address this limitation, the concepts of *HyperGraphs* and *SuperHyperGraphs* have been introduced and actively studied. A *hypergraph* generalizes a simple graph by allowing each edge—called a *hyperedge*—to link any nonempty subset of vertices, making it possible to model interactions involving more than two entities [1–5]. HyperGraphs have been explored in diverse application areas, including graph neural networks [6–10], decision-making [11, 12], and chemistry [13, 14].

A *superhypergraph* builds upon the hypergraph framework by recursively applying the powerset operation, resulting in hierarchically nested hyperedges and enabling the representation of multi-level relationships. This broader formalism has been analyzed in a number of recent works [15–20]. SuperHyperGraphs have likewise attracted interest in various domains such as graph neural networks [21–23], engineering [24–27], and decision-making [18, 28, 29], where their ability to express complex interdependencies is especially valuable.

Definition 1.1 (Base Set)**.** [30] Let S be a nonempty set, called the *base set*. All higher-order objects, such as powersets and supervertices, are constructed from S:

$$S = \{ x \mid x \text{ is an element of the domain} \}.$$

Definition 1.2 (Powerset)**.** (cf. [31, 32]) For any set S, its *powerset* $\mathrm{POWS}(S)$ is the collection of all subsets of S, including $\emptyset$ and S itself:

$$\mathrm{POWS}(S) = \{ A \mid A \subseteq S \}.$$

Definition 1.3 (n-th Powerset)**.** [19, 33–35]. Let H be a set. The *n-th powerset* $\mathrm{POWS}^n(H)$ is defined recursively by

$$\mathrm{POWS}^0(H) = H, \qquad \mathrm{POWS}^{k+1}(H) = \mathrm{POWS}\big(\mathrm{POWS}^k(H)\big), \quad k \geq 0.$$

The *nonempty n-th powerset* $\mathrm{POWS}^{*n}(H)$ is defined similarly but removes the empty set at each stage:

$$\mathrm{POWS}^{*0}(H) = H, \qquad \mathrm{POWS}^{*(k+1)}(H) = \mathrm{POWS}^*\big(\mathrm{POWS}^{*k}(H)\big),$$

where $\mathrm{POWS}^*(X) = \mathrm{POWS}(X) \setminus \{\emptyset\}$

Example 1.4 (n-th powerset in everyday planning: nested menu design)**.** Let $H = \{\text{salad}, \text{soup}, \text{pasta}\}$ be the set of available dishes. Then $\mathrm{POWS}^1(H)$ lists all possible menus (any subset of dishes). An element of the second powerset,

$$M = \big\{\{\text{salad}\},\ \{\text{soup}, \text{pasta}\}\big\} \in \mathrm{POWS}^2(H),$$

may be read as a *menu of menus*: the caterer offers either the single–dish menu $\{\text{salad}\}$ or the two–dish menu $\{\text{soup}, \text{pasta}\}$. Likewise, a collection $C = \{M,\ M'\} \in \mathrm{POWS}^3(H)$ encodes a *catalog* of alternative menu packages. This illustrates how POWS^n captures nested choice structures (choices, choices of choices, etc.).

Definition 1.5 (HyperGraph)**.** [36, 37] A finite *HyperGraph* $H = (V, E)$ is specified by:

- A nonempty vertex set V.
- A collection E of hyperedges, where each $e \in E$ is a nonempty subset of V.

HyperGraphs generalize ordinary graphs by permitting edges to join any number of vertices, making them ideal for modeling higher-order relationships. In this work, we assume both V and E are finite.

Example 1.6 (Hypergraph in practice: coauthorship network)**.** Let $V = \{\text{Rika}, \text{Yuya}, \text{Chen}, \text{Dee}\}$ be a set of researchers. Define hyperedges by papers:

$$E = \big\{\{\text{Rika}, \text{Yuya}\},\ \{\text{Yuya}, \text{Chen}, \text{Dee}\},\ \{\text{Rika}, \text{Chen}\}\big\}.$$

Then $H = (V, E)$ is a hypergraph where each paper (hyperedge) connects *all* of its coauthors simultaneously, naturally representing a group collaboration that cannot be reduced to pairwise links without loss of information.

Definition 1.7 (Level-n SuperHyperGraph (incidence form))**.** (cf. [16, 19, 23, 38]) Fix a finite base set V_0 and an integer $n \geq 0$. Let $V_n \subseteq \mathrm{POWS}^n(V_0)$ be a finite set, whose elements are called *n-supervertices*. A *level-n SuperHyperGraph* is a pair

$$\mathcal{H}^{(n)} = (V_n, \mathcal{E}), \quad \text{with} \quad \emptyset \neq \mathcal{E} \subseteq \mathrm{POWS}(V_n) \setminus \{\emptyset\}.$$

Thus each *n-superedge* $E \in \mathcal{E}$ is a nonempty subset of the vertex set V_n. (When $n = 0$, this is an ordinary finite hypergraph; when, additionally, every $E \in \mathcal{E}$ has size 2, it is an ordinary graph.)

Example 1.8 (Level-2 SuperHyperGraph in organizations: programs of teams)**.** Let the base set of people be $V_0 = \{a, b, c, d, e\}$. Form teams (level 1 supervertices)

$$T_1 = \{a, b\}, \quad T_2 = \{c, d, e\}, \quad T_3 = \{a, c\},$$

and set $V_1 = \{T_1, T_2, T_3\} \subseteq \mathrm{POWS}^1(V_0)$. Create programs (level 2 supervertices) as collections of teams

$$P_1 = \{T_1, T_2\}, \qquad P_2 = \{T_2, T_3\},$$

and let $V_2 = \{P_1, P_2\} \subseteq \mathrm{POWS}^2(V_0)$. Define a level-2 SuperHyperGraph $\mathcal{H}^{(2)} = (V_2, \mathcal{E})$ with superedges

$$\mathcal{E} = \big\{ \{P_1, P_2\}, \{P_1\} \big\} \subseteq \mathrm{POWS}(V_2) \setminus \{\emptyset\}.$$

Here, $\{P_1, P_2\}$ models a joint milestone linking both programs, while the singleton $\{P_1\}$ models an internal review of P_1 alone. This realizes the incidence-form definition: level-2 supervertices are elements of $\mathrm{POWS}^2(V_0)$, and superedges are nonempty subsets of V_2.

Notation 1.9 (Stars)**.** *For $\mathcal{H}^{(n)} = (V_n, \mathcal{E})$ and $v \in V_n$, the* star *of v is*

$$\mathsf{Star}_{\mathcal{H}}(v) := \{ E \in \mathcal{E} : v \in E \} \subseteq \mathcal{E}.$$

We also write $\mathcal{E}^{\neq\emptyset}(v) := \mathsf{Star}_{\mathcal{H}}(v)$ and $\mathcal{E}^{(\geq 2)}(v) := \{ E \in \mathcal{E} : v \in E \text{ and } |E| \geq 2 \}$ when we wish to exclude size-1 edges in the star.

1.2 directed hypergraph

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. [39–48]). Its definition is provided below.

Definition 1.10 (Directed Graph)**.** [49] 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).

Example 1.11 (Directed Graph: One–Way Urban Road Network)**.** Let V be the set of road intersections in a downtown district and let $E \subseteq V \times V$ contain an ordered pair (u, v) whenever there is a one–way street that allows travel from intersection u to intersection v. Then $G = (V, E)$ is a directed graph (digraph). For instance, with $V = \{A, B, C, D\}$ and one–way streets $A \to B$, $B \to C$, $A \to D$, $D \to C$, we have

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

Shortest–path queries model quickest driving routes; in– and out–degrees describe traffic inflow/outflow at each intersection; cycles (e.g., $A \to B \to C \to A$) identify potential circulation patterns.

Definition 1.12 (Directed Hypergraph)**.** [50, 51] 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.

Example 1.13 (Directed Hypergraph: Course Prerequisite System)**.** Let V be the set of courses at a university (each element is a single course). Define a directed hypergraph $H = (V, E)$ where a hyperarc $e = (\mathrm{Tail}(e), \mathrm{Head}(e))$ encodes a prerequisite rule that *all* courses in the tail must be completed before enrolling in the head course. For example, with

$$V = \{\mathrm{Math101},\ \mathrm{Phys101},\ \mathrm{CS101},\ \mathrm{Data201},\ \mathrm{AI301}\},$$

the rules

$$(\{\mathrm{Math101}, \mathrm{CS101}\},\ \mathrm{Data201}), \qquad (\{\mathrm{Data201}, \mathrm{Phys101}\},\ \mathrm{AI301})$$

mean that *Data201* requires both *Math101* and *CS101*, while *AI301* requires both *Data201* and *Phys101*. Here each tail is a subset of V and the head is a single course in V, exactly matching the directed-hypergraph definition.

Define the *Directed n-SuperHypergraph* and examine its relationships with other graph classes. The definition and basic properties are given below.

Definition 1.14 (Directed n-SuperHypergraph)**.** [38, 52] Let V be a finite, nonempty set of vertices, and let the iterated powersets be defined recursively by

$$\mathrm{POWS}^0(V) = V, \qquad \mathrm{POWS}^{k+1}(V) = \mathrm{POWS}\big(\mathrm{POWS}^k(V)\big) \quad (k \geq 0).$$

A *Directed n-SuperHypergraph* is a tuple

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

where $E \subseteq \mathrm{POWS}^n(V) \times \mathrm{POWS}^n(V)$ is a set of *directed n-superhyperedges*. Each $e = (T, H) \in E$ consists of

- $T \in \mathrm{POWS}^n(V)$: the **tail object** (a nested "source" collection), and
- $H \in \mathrm{POWS}^n(V)$: the **head object** (a nested "target" collection).

When convenient, we require $T, H \neq \emptyset$ for $n \geq 1$ (nonemptiness of the outer level).

Remark 1.15.

- For $n = 0$, one has $\mathrm{POWS}^0(V) = V$, so $E \subseteq V \times V$; thus DSH_0 is an ordinary directed graph (digraph).
- For $n = 1$, one has $E \subseteq \mathrm{POWS}(V) \times \mathrm{POWS}(V)$; hence DSH_1 is a directed hypergraph (tails/heads are vertex *sets*).
- For $n \geq 2$, T, H are *nested* collections (sets of sets, etc.), enabling multi-level grouping of sources and targets and capturing hierarchical/aggregated interactions.
- If $T = \{v_1\}$ and $H = \{v_2\}$ at $n = 0$ (or $T = \{\{v_1\}\}, H = \{\{v_2\}\}$ at $n = 2$, etc., via singleton lifts), the edge behaves like a standard directed edge $v_1 \longrightarrow v_2$.

Example 1.16 (Directed n-SuperHypergraph ($n = 2$): Multi–Team Approval Workflow)**.** Let V be the set of individual staff members in an organization. Level–1 collections are *teams* (nonempty subsets of V), i.e., elements of $\mathrm{POWS}^1(V) = \mathrm{POWS}(V)$. Level–2 collections are *consortia of teams*, i.e., elements of $\mathrm{POWS}^2(V) = \mathrm{POWS}(\mathrm{POWS}(V))$. Define a directed 2–superhypergraph $\mathrm{DSH}_2 = (V, E)$ with edges $E \subseteq \mathrm{POWS}^2(V) \times \mathrm{POWS}^2(V)$ as follows. Suppose three teams are

$$G_1 = \{a, b\}, \qquad G_2 = \{c\}, \qquad G_3 = \{d, e, f\} \subset V,$$

and consider the consortia

$$C_{\mathrm{approve}} = \{\, G_1, G_2 \,\} \in \mathrm{POWS}^2(V), \qquad C_{\mathrm{handoff}} = \{\, G_3 \,\} \in \mathrm{POWS}^2(V).$$

Create a directed 2–superhyperedge

$$e = \big(C_{\text{approve}} \,,\, C_{\text{handoff}} \big) \;\in\; \mathrm{POWS}^2(V) \times \mathrm{POWS}^2(V).$$

This models the real–life rule: "An approval may be issued by *either* team G_1 or team G_2 (the level–2 tail object is a set of teams), after which the work is handed to team G_3 (the level–2 head object is the singleton set $\{G_3\}$)." Here both tail and head are elements of $\mathrm{POWS}^2(V)$, so the construction is a concrete instance of a directed n–superhypergraph with $n = 2$.

Theorem 1.17 (Reduction ladder)**.** *Let* $\mathrm{DSH}_n = (V, E)$ *be a Directed n-SuperHypergraph.*

1. *If* $n = 0$*,* DSH_0 *is exactly a directed graph.*
2. *If* $n = 1$*,* DSH_1 *is exactly a directed hypergraph.*
3. *For any* $n \geq 1$*, restricting to* singleton-lifted *edges*

$$E' \;=\; \Big\{ \big(\{S\}, \{T\}\big) \in E \,\Big|\, S, T \in \mathrm{POWS}^{n-1}(V) \Big\}$$

yields a Directed $(n-1)$*-SuperHypergraph on V (after identifying* $\{S\} \mapsto S$*). Iterating this identification reduces any* DSH_n *to a directed hypergraph (*n=1*), and further to a digraph (*n=0*) when tails/heads are singletons.*

Proof. (1) and (2) are immediate from $\mathrm{POWS}^0(V) = V$ and $\mathrm{POWS}^1(V) = \mathrm{POWS}(V)$. For (3), observe that if every tail/head at level n is a singleton $\{S\}$ with $S \in \mathrm{POWS}^{n-1}(V)$, then the map $\{S\} \mapsto S$ is a bijection between the allowed tails/heads and $\mathrm{POWS}^{n-1}(V)$. Thus E' corresponds to a subset of $\mathrm{POWS}^{n-1}(V) \times \mathrm{POWS}^{n-1}(V)$, i.e., a Directed $(n-1)$-SuperHypergraph. Repeating the argument establishes the stated reductions. □

2 Line Directed HyperGraph

2.1 Line Directed Graph

A line graph represents adjacencies between edges of a graph, forming vertices connected when edges share an endpoint [53–55]. Line directed graphs transform a directed graph into another whose vertices represent edges, preserving directed adjacency relationships (cf. [45, 56–59]).

Definition 2.1 (Line Directed Graph)**.** (cf. [58]) Let $D = (V, A)$ be a finite directed graph (*digraph*), where V is the vertex set and $A \subseteq V \times V$ is the arc set. The *line directed graph* of D, denoted $L(D)$, is the digraph whose

- vertices are the arcs of D, i.e., $V\big(L(D)\big) = A$;
- there is an arc in $L(D)$ from $e_1 = (u, v) \in A$ to $e_2 = (x, y) \in A$ if and only if $v = x$, i.e., the head of e_1 equals the tail of e_2:

$$A\big(L(D)\big) = \{ \big((u, v), (v, y)\big) \in A \times A \;\mid\; (u, v) \in A,\; (v, y) \in A \}.$$

This is the standard definition: "$L(D)$ has the arcs of D as vertices; there is an arc from pq to uv iff $q = u$.

2.2 Line Directed HyperGraph

Line directed hypergraphs map a directed hypergraph into a hypergraph where vertices represent hyperedges, maintaining directed incidence structure.

Definition 2.2 (Line Directed HyperGraph)**.** Let $H = (V, E)$ be a directed hypergraph. Its *Line Directed HyperGraph* is the directed hypergraph

$$\mathrm{LDHG}(H) := (E,\ E^{\mathrm{line}})$$

whose vertex set is the hyperarc set E of H, and whose hyperarc set is

$$E^{\mathrm{line}} := \Big\{(\{e_1\},\ e_2)\ \Big|\ e_1, e_2 \in E,\ \mathrm{Head}(e_1) \in \mathrm{Tail}(e_2)\Big\}.$$

Thus, for every admissible composable pair $e_1 \to e_2$ in H (i.e. the head of e_1 belongs to the tail of e_2), we place in $\mathrm{LDHG}(H)$ a (singleton–tail) hyperarc from the vertex e_1 to the vertex e_2.

Example 2.3 (Line Directed HyperGraph: Manufacturing Flow of a PCB)**.** Consider a directed hypergraph $H = (V, E)$ whose vertices V are *artifacts* in a printed–circuit–board (PCB) production line and whose hyperarcs encode *process steps*: the tail lists all required inputs and the head is the produced output. Let

$$V = \{\text{RawBoard, SolderPaste, SolderedBoard, Components, AssembledBoard, TestFixture, TestedBoard}\},$$

and define the process steps (hyperarcs)

$$\begin{aligned} e_1 &= (\{\text{RawBoard}, \text{SolderPaste}\},\ \text{SolderedBoard}),\\ e_2 &= (\{\text{SolderedBoard}, \text{Components}\},\ \text{AssembledBoard}),\\ e_3 &= (\{\text{AssembledBoard}, \text{TestFixture}\},\ \text{TestedBoard}). \end{aligned}$$

In the *line directed hypergraph* of H, the vertices are the process steps $\{e_1, e_2, e_3\}$, and we add a (singleton–tail) hyperarc from e_i to e_j exactly when the *output* of e_i (its head) is among the *inputs* of e_j (its tail). Hence

$$E^{\mathrm{line}} = \big\{\ (\{e_1\}, e_2),\ (\{e_2\}, e_3)\ \big\},$$

because $\mathrm{Head}(e_1) = \text{SolderedBoard} \in \mathrm{Tail}(e_2)$ and $\mathrm{Head}(e_2) = \text{AssembledBoard} \in \mathrm{Tail}(e_3)$. The resulting line directed hypergraph compactly captures which manufacturing steps can follow which others.

Theorem 2.4 (LDHG is a directed hypergraph and generalizes the line digraph)**.** *Let $H = (V, E)$ be a directed hypergraph and let* $\mathrm{LDHG}(H) = (E, E^{\mathrm{line}})$ *be as in Definition 2.2. Then:*

(a) ***(Well-defined directed hypergraph).*** $\mathrm{LDHG}(H)$ *is a directed hypergraph: every element of E^{line} has a nonempty tail (a singleton $\{e_1\}$ with $e_1 \in E$) and a head in the vertex set E.*

(b) ***(Reduction to the line directed graph).*** *Suppose H comes from a digraph $D = (V, A)$ by identifying each arc $a = (u, v) \in A$ with the hyperarc $e_a = (\{u\}, v) \in E$. Then* $\mathrm{LDHG}(H)$ *is canonically isomorphic to the classical line directed graph $L(D)$:*

$$V\big(\mathrm{LDHG}(H)\big) = A, \qquad \big(\{(u,v)\}, (v,w)\big) \in E^{\mathrm{line}} \iff \big((u,v),(v,w)\big) \in A\big(L(D)\big).$$

In particular, LDHG *is a strict generalization of the line directed graph.*

Proof. (a) By construction, vertices of $\mathrm{LDHG}(H)$ are the elements of E. Given $e_1, e_2 \in E$ with $\mathrm{Head}(e_1) \in \mathrm{Tail}(e_2)$, the pair $\big(\{e_1\}, e_2\big)$ has a nonempty tail $\{e_1\} \subseteq E$ and its head e_2 is a vertex of $\mathrm{LDHG}(H)$; hence it is a valid hyperarc. Therefore $E^{\mathrm{line}} \subseteq \{\ (\emptyset \neq S \subseteq E,\ \mathrm{head} \in E)\ \}$, so $\mathrm{LDHG}(H)$ is a directed hypergraph.

(b) Let $D = (V, A)$ be a digraph and form $H = (V, E)$ with $E = \{\ (\{u\}, v) \mid (u,v) \in A\ \}$. Define a bijection $\varphi : A \to V\big(\mathrm{LDHG}(H)\big)$ by $\varphi\big((u,v)\big) = (\{u\}, v)$. For arcs, we have

$$\big(\{(u,v)\}, (v,w)\big) \in E^{\mathrm{line}} \quad\iff\quad \mathrm{Head}\big((\{u\}, v)\big) = v\ \in\ \mathrm{Tail}\big((\{v\}, w)\big) = \{v\},$$

which holds exactly when $v = v$, i.e. always for the composable pair $(u,v) \to (v,w)$. Thus $\big(\{(u,v)\}, (v,w)\big) \in E^{\mathrm{line}}$ if and only if $\big((u,v),(v,w)\big) \in A\big(L(D)\big)$. Transporting arcs through φ identifies $\mathrm{LDHG}(H)$ with $L(D)$ (note that every tail in $\mathrm{LDHG}(H)$ is a singleton, hence $\mathrm{LDHG}(H)$ is a digraph in this special case). Therefore the construction reduces to the classical line directed graph, proving the claim. □

2.3 Line Directed SuperHyperGraph

Line directed superhypergraphs convert a directed superhypergraph into one whose vertices are superhyperedges, preserving hierarchical directed connections.

Definition 2.5 (Line Directed SuperHyperGraph)**.** Let $\mathrm{DSH}_n = (V, E)$ be a Directed n-SuperHypergraph. Its *Line Directed SuperHyperGraph* is the Directed 1-SuperHypergraph

$$\mathrm{LDSH}(\mathrm{DSH}_n) \;:=\; \big(E,\; E^{\mathrm{line}}\big),$$

with base set E (the n-superhyperedges of the original object) and

$$E^{\mathrm{line}} \;:=\; \Big\{ \big(\{e_1\},\, \{e_2\}\big) \in \mathrm{POWS}(E) \times \mathrm{POWS}(E) \;\Big|\; e_1 = (T_1, H_1),\; e_2 = (T_2, H_2) \in E,\; H_1 \subseteq T_2 \Big\}.$$

That is, there is a *line superhyperedge* from the vertex e_1 to the vertex e_2 (we write $e_1 \longrightarrow e_2$) precisely when the head object of e_1 is contained in the tail object of e_2 *at level* n, i.e. $H_1 \subseteq T_2$ as elements of $\mathrm{POWS}^n(V)$.

Example 2.6 (Line Directed SuperHyperGraph: Multi–Team Approval and Handoff)**.** Let V be the set of *individuals* in an organization. Level–1 objects $\mathrm{POWS}(V) \setminus \{\emptyset\}$ represent *teams* (nonempty subsets of individuals). Level–2 objects $\mathrm{POWS}(\mathrm{POWS}(V))$ represent *consortia of teams*. Consider a directed 2–superhypergraph $\mathrm{DSH}_2 = (V, E)$ whose 2–superhyperedges encode policies: the tail is a set of teams authorized to perform an action; the head is the set of teams to which the work is handed off.

Let three teams be

$$G_{\mathrm{Design}} = \{a, b\}, \quad G_{\mathrm{QA}} = \{c\}, \quad G_{\mathrm{Mfg}} = \{d\} \;\subset V.$$

Define two 2–superhyperedges (tail/head are level–2 objects—sets of teams):

$$\begin{aligned} e_A &= \big(\,\{\,G_{\mathrm{Design}}\,\},\; \{\,G_{\mathrm{QA}}\,\}\,\big) && \text{(design completion hands off to QA),} \\ e_B &= \big(\,\{\,G_{\mathrm{QA}},\; G_{\mathrm{Mfg}}\,\},\; \{\,G_{\mathrm{Mfg}}\,\}\,\big) && \text{(QA or Mfg may initiate, then hand off within Mfg).} \end{aligned}$$

Since the head object of e_A is $\{G_{\mathrm{QA}}\}$ and this is contained in the tail object of e_B, i.e. $\{G_{\mathrm{QA}}\} \subseteq \{G_{\mathrm{QA}}, G_{\mathrm{Mfg}}\}$, the *line directed superhypergraph* has a line superhyperedge from e_A to e_B:

$$(\;\{e_A\},\; \{e_B\}\;) \;\in\; E^{\mathrm{line}}.$$

Thus, vertices are the policy edges $\{e_A, e_B\}$, and a line superhyperedge records a valid *composition of policies*: whatever consortium receives the handoff from e_A is admissible as part of the initiating consortium for e_B.

Theorem 2.7 (LDSH is a Directed SuperHyperGraph)**.** *For every Directed n-SuperHypergraph $\mathrm{DSH}_n = (V, E)$, the construction $\mathrm{LDSH}(\mathrm{DSH}_n) = (E, E^{\mathrm{line}})$ of Definition 2.5 is a Directed 1-SuperHypergraph on the base set E.*

Proof. By definition $E^{\mathrm{line}} \subseteq \mathrm{POWS}(E) \times \mathrm{POWS}(E)$, so edges of $\mathrm{LDSH}(\mathrm{DSH}_n)$ are pairs of level-1 objects over E (singleton tails and heads). Thus $\mathrm{LDSH}(\mathrm{DSH}_n)$ is a Directed 1-SuperHypergraph on base set E. □

Theorem 2.8 (Generalizes the line directed graph)**.** *Let $D = (V, A)$ be a finite digraph and consider the Directed 1-SuperHypergraph $\mathrm{DSH}_1(D) = (V, E)$ obtained by identifying each arc $(u, v) \in A$ with the superhyperedge $e_{(u,v)} = (\{u\}, \{v\}) \in \mathrm{POWS}(V) \times \mathrm{POWS}(V)$. Then $\mathrm{LDSH}\big(\mathrm{DSH}_1(D)\big)$ is canonically isomorphic to the classical line directed graph $L(D)$.*

Proof. Vertices of LDSH are $E = \{e_{(u,v)} : (u, v) \in A\}$; identify $e_{(u,v)}$ with the arc (u, v). By Definition 2.5, a line edge from $e_{(u,v)}$ to $e_{(x,y)}$ exists iff $H_1 = \{v\} \subseteq T_2 = \{x\}$, i.e. $v = x$. Thus there is an arc $(u, v) \longrightarrow (x, y)$ in LDSH exactly when there is the standard line-graph arc in $L(D)$. This gives a canonical isomorphism. □

Theorem 2.9 (Generalizes the line directed hypergraph)**.** *Let $H = (V, E)$ be a directed hypergraph in the single-head convention ($E \subseteq \{(T, h) : \emptyset \neq T \subseteq V,\; h \in V\}$). Embed H as a Directed 1-SuperHypergraph $\mathrm{DSH}_1(H) = (V, \widehat{E})$ by*

$$(T, h) \;\longmapsto\; \widehat{e} = (T, \{h\}) \in \mathrm{POWS}(V) \times \mathrm{POWS}(V).$$

Then $\mathrm{LDSH}\big(\mathrm{DSH}_1(H)\big)$ agrees with the usual Line Directed HyperGraph (after the obvious identification of a singleton head $\{\,\widehat{e}_2\,\}$ with the vertex $\widehat{e}_2$).

Proof. Vertices of LDSH are the (embedded) hyperarcs $\widehat{e} = (T, \{h\})$ of H. By Definition 2.5, there is a line edge from $\widehat{e}_1 = (T_1, \{h_1\})$ to $\widehat{e}_2 = (T_2, \{h_2\})$ iff $\{h_1\} \subseteq T_2$, i.e. $h_1 \in T_2$. This is exactly the standard composability condition used in the line directed hypergraph: the head of the first hyperarc lies in the tail of the second. If one identifies the singleton head $\{ \widehat{e}_2 \}$ with the vertex $\widehat{e}_2$, the edge set coincides with that of the usual LDHG. $\square$

3 Iterated line directed graphs

3.1 Iterated line directed graphs

Iterated line directed graphs repeatedly apply the line graph operation to a directed graph, preserving directional adjacency structure [60–66].

Definition 3.1 (Iterated line digraph)**.** [67, 68] Define $L^0(G) := G$ and, recursively, $L^{k+1}(G) := L\big(L^k(G)\big)$ for each integer $k \geq 0$. We call $L^k(G)$ the *k-iterated line digraph* of G.

Example 3.2 (Iterated line digraph of a short path)**.** Let $G = (V, A)$ be the directed path $1 \to 2 \to 3$, i.e.,

$$V = \{1, 2, 3\}, \qquad A = \{(1, 2), (2, 3)\}.$$

Then the line digraph $L(G)$ has

$$V\big(L(G)\big) = A = \{e_{12}, e_{23}\}, \quad A\big(L(G)\big) = \{(e_{12}, e_{23})\}$$

since the head of e_{12} equals the tail of e_{23}. Iterating once more, $L^2(G) = L\big(L(G)\big)$ has

$$V\big(L^2(G)\big) = A\big(L(G)\big) = \{(e_{12}, e_{23})\}, \qquad A\big(L^2(G)\big) = \emptyset.$$

Thus $L(G)$ is a single directed edge, and $L^2(G)$ is a single isolated vertex.

Proposition 3.3 (Walk model and adjacency in $L^k(G)$)**.** *For every $k \geq 1$:*

(i) The vertices of $L^k(G)$ are in one-to-one correspondence with directed walks

$$(v_0, v_1, \ldots, v_k) \quad \textit{in } G, \;\; \textit{i.e.}, \; (v_{i-1}, v_i) \in A(G) \textit{ for } i = 1, \ldots, k.$$

(ii) Under this correspondence, there is an arc in $L^k(G)$ from the walk $(v_0, \ldots, v_k)$ to the walk $(v_1, \ldots, v_k, v_{k+1})$ for each $(v_k, v_{k+1}) \in A(G)$.

Proof. The statement is immediate for $k = 1$ from the definition of $L(G)$. For $k \geq 2$, a vertex of $L^k(G) = L\big(L^{k-1}(G)\big)$ is an arc of $L^{k-1}(G)$, hence a pair of consecutive vertices of $L^{k-1}(G)$. By the induction hypothesis, vertices of $L^{k-1}(G)$ correspond to length-$(k-1)$ walks in G, and an arc of $L^{k-1}(G)$ corresponds to sliding a length-$(k-1)$ window by one step in G. Therefore vertices of $L^k(G)$ correspond to length-k walks, and adjacency corresponds to the one-step shift $(v_0, \ldots, v_k) \to (v_1, \ldots, v_k, v_{k+1})$ whenever $(v_k, v_{k+1}) \in A(G)$. $\square$

3.2 Iterated line directed hypergraphs

Iterated line directed hypergraphs repeatedly form line hypergraphs from a directed hypergraph, maintaining directed incidence between hyperedges.

Definition 3.4 (Iterated line directed hypergraphs)**.** Let $H = (V, E)$ be a directed hypergraph. Define

$$L^1(H) := L(H), \qquad L^{k+1}(H) := L\big(L^k(H)\big) \quad (k \geq 1).$$

We call $L^k(H)$ the *k-iterated line directed hypergraph* of H.

Example 3.5 (Iterated line directed hypergraphs: a small production workflow)**.** Consider a (finite) directed hypergraph $H = (V, E)$ that models a toy food production line. The vertices are ingredients and intermediates

$$V = \{\text{Flour}, \text{Water}, \text{Yeast}, \text{Dough}, \text{Sauce}, \text{Cheese}, \text{PizzaBase}, \text{Pizza}\}.$$

Hyperarcs encode *operations* that consume several inputs (tail) to produce one output (head):

$$\begin{aligned} e_1 &: \ (\{\text{Flour}, \text{Water}, \text{Yeast}\},\ \text{Dough}), \\ e_2 &: \ (\{\text{Dough}\},\ \text{PizzaBase}), \\ e_3 &: \ (\{\text{Sauce}, \text{Cheese}, \text{PizzaBase}\},\ \text{Pizza}). \end{aligned}$$

The line directed hypergraph $L(H)$ has the *operations* e_1, e_2, e_3 as its vertices. There is a (singleton–tail) hyperarc from e_i to e_j exactly when the head of e_i is an element of the tail of e_j. Hence

$$(\{e_1\}, e_2) \in E(L(H)) \quad \text{since Dough} \in \{\text{Dough}\},$$

$$(\{e_2\}, e_3) \in E(L(H)) \quad \text{since PizzaBase} \in \{\text{Sauce}, \text{Cheese}, \text{PizzaBase}\}.$$

There is no hyperarc $(\{e_1\}, e_3)$, because Dough $\notin \{\text{Sauce}, \text{Cheese}, \text{PizzaBase}\}$.

Iterating once more, the second line hypergraph $L^2(H) = L(L(H))$ has as vertices the hyperarcs of $L(H)$, i.e. $\{(\{e_1\}, e_2), (\{e_2\}, e_3)\}$. Since the head of $(\{e_1\}, e_2)$ equals e_2 and the tail of $(\{e_2\}, e_3)$ is $\{e_2\}$, we obtain

$$\big(\ \{(\{e_1\}, e_2)\}\ ,\ (\{e_2\}, e_3)\ \big) \ \in\ E\big(L^2(H)\big),$$

which compactly represents the two–step operation chain $e_1 \to e_2 \to e_3$ ("make dough → press base → bake pizza"). In general, $L^k(H)$ encodes directed chains of k composable operations as vertices, with adjacency corresponding to a one–step shift along such chains.

Remark 3.6. For $k \geq 1$ define the set of *compatible k-sequences of hyperarcs*

$$\mathsf{W}_k(H) \ :=\ \Big\{(e_1, \ldots, e_k) \in E^k \ \Big|\ h(e_i) \in T(e_{i+1}) \text{ for all } i = 1, \ldots, k-1\Big\}.$$

Intuitively, $(e_1, \ldots, e_k)$ is a directed *walk of hyperarcs* in which the head of each step belongs to the next tail.

Theorem 3.7 (Basic properties and walk correspondence)**.** *Let $H = (V, E)$ be a directed hypergraph.*

(a) *$L(H)$ is a directed hypergraph in the sense above. Moreover, by Definition, all hyperarcs of $L(H)$ have singleton tails and single-vertex heads:*

$$E\big(L(H)\big) \ \subseteq\ \big\{\ (\{x\}, y)\ :\ x, y \in V\big(L(H)\big)\ \big\}.$$

(b) *For every $k \geq 1$, $L^k(H)$ is a directed hypergraph and there is a natural bijection*

$$\iota_k :\ \mathsf{W}_k(H) \ \xrightarrow{\cong}\ V\big(L^k(H)\big)$$

satisfying the recursion

$$\iota_1(e_1) \ :=\ e_1, \qquad \iota_{k+1}(e_1, \ldots, e_{k+1}) \ :=\ \big(\ \{\ \iota_k(e_1, \ldots, e_k)\ \}\ ,\ \iota_k(e_2, \ldots, e_{k+1})\ \big).$$

Under this identification, adjacency in $L^k(H)$ is the one-step shift: there is a hyperarc

$$\big(\ \{\ \iota_k(e_1, \ldots, e_k)\ \}\ ,\ \iota_k(e_2, \ldots, e_{k+1})\ \big) \ \in\ E\big(L^k(H)\big)$$

for each $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}(H)$.

Proof. (a) By Definition, $V(L(H)) = E$ and each listed pair $\big(\{e\}, f\big)$ has a nonempty tail $\{e\} \subseteq V(L(H))$ and a head $f \in V(L(H))$, so $L(H)$ is indeed a directed hypergraph.

(b) We proceed by induction on k. For $k = 1$ the statement is immediate: $V(L(H)) = E = \mathsf{W}_1(H)$ via $\iota_1(e_1) = e_1$.

Assume the claim holds for k. Then $L^{k+1}(H) = L(L^k(H))$ has vertex set $V(L^{k+1}(H)) = E(L^k(H))$. By the definition of the line operator, a hyperarc of $L^k(H)$ is any pair

$$\Big(\{\ u\ \},\ v \Big) \quad \text{with } u, v \in V(L^k(H)) \text{ and } \ h_{L^k}(u) \in T_{L^k}(v).$$

By the induction hypothesis, every $u \in V(L^k(H))$ is of the form $u = \iota_k(e_1, \ldots, e_k)$ and every $v \in V(L^k(H))$ is of the form $v = \iota_k(e_2, \ldots, e_{k+1})$ for some $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}(H)$. Moreover, by construction of ι_k one checks directly that

$$h_{L^k}\big(\iota_k(e_1, \ldots, e_k)\big) \;=\; \iota_{k-1}(e_2, \ldots, e_k), \qquad T_{L^k}\big(\iota_k(e_2, \ldots, e_{k+1})\big) \;=\; \big\{\iota_{k-1}(e_2, \ldots, e_k)\big\}.$$

Hence the line condition $h_{L^k}(u) \in T_{L^k}(v)$ is satisfied *iff* $u = \iota_k(e_1, \ldots, e_k)$ and $v = \iota_k(e_2, \ldots, e_{k+1})$ for a compatible sequence, i.e., $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}(H)$. This shows that

$$\iota_{k+1}(e_1, \ldots, e_{k+1}) \;:=\; \big(\{\iota_k(e_1, \ldots, e_k)\},\ \iota_k(e_2, \ldots, e_{k+1})\big)$$

parametrizes all vertices of $L^{k+1}(H)$ bijectively, and the displayed hyperarc is present *exactly* for $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}(H)$. Thus the adjacency in $L^{k+1}(H)$ is the stated one-step shift, completing the induction. □

Theorem 3.8 (Generalization ladder)**.** *Let $H = (V, E)$ be a directed hypergraph.*

(A) ***(k=1 gives the line directed hypergraph).*** *For $k = 1$, $L^1(H) = L(H)$ is exactly the line directed hypergraph of H.*

(B) ***(Reduction to iterated line digraphs).*** *Suppose H comes from a digraph $D = (V, A)$ by replacing each arc $(u, v) \in A$ with the hyperarc $e_{uv} = \big(\{u\}, v\big)$. Then, for every $k \geq 1$, there is a natural digraph isomorphism*

$$L^k(H) \;\cong\; L^k(D),$$

obtained by identifying a vertex $\iota_k(e_{v_0v_1}, \ldots, e_{v_{k-1}v_k}) \in V(L^k(H))$ with the directed walk $(v_0, \ldots, v_k) \in V\big(L^k(D)\big)$.

(C) ***(LDH generalizes line digraphs).*** *In particular, when all hyperarcs of H have singleton tails ($|T(e)| = 1$) and distinct heads, $L(H)$ coincides with the classical line digraph on the arc set: there is an arc from e_{uv} to e_{vw} in $L(H)$ iff v is the common endpoint, i.e., iff $h(e_{uv}) = v \in T(e_{vw}) = \{v\}$.*

Proof. (A) is immediate from the definition $L^1(H) := L(H)$.

For (B), by construction $V(L(H)) = E = \{e_{uv} : (u, v) \in A\}$ and

$$\big(\{e_{uv}\},\ e_{vw}\big) \in E(L(H)) \quad\Longleftrightarrow\quad h(e_{uv}) = v \in T(e_{vw}) = \{v\} \quad\Longleftrightarrow\quad (u, v) \to (v, w) \text{ in } D,$$

so $L(H) \cong L(D)$. Iterating and using Theorem 3.7(b), the vertices of $L^k(H)$ are in bijection with compatible sequences $(e_{v_0v_1}, \ldots, e_{v_{k-1}v_k})$, equivalently with directed walks $(v_0, \ldots, v_k)$ in D, and adjacency is the one-step shift on walks in both constructions. Hence $L^k(H) \cong L^k(D)$.

For (C), take $k = 1$ in (B). The displayed equivalence shows $L(H)$ agrees with the usual line digraph when every $T(e)$ is a singleton, as claimed. □

3.3 Iterated line directed superhypergraphs

Iterated line directed superhypergraphs repeatedly build line superhypergraphs from a directed superhypergraph, preserving hierarchical directed incidence patterns.

Remark 3.9. For $n \geq 0$ define the *compatibility* predicate C_n between $X, Y \in \mathrm{POWS}^n(V)$ by

$$C_0(x, y) \;\Leftrightarrow\; x = y \quad (x, y \in V), \qquad C_n(X, Y) \;\Leftrightarrow\; X \cap Y \neq \emptyset \quad (n \geq 1).$$

For a set W and $x \in W$, define the *n-fold singleton lift* $\sigma_0(x) := x$ and $\sigma_{k+1}(x) := \{\sigma_k(x)\}$, so that $\sigma_n(x) \in \mathrm{POWS}^n(W)$ is the n-times nested singleton of x.

For $k \geq 1$, define the set of *n-compatible k-walks of superhyperedges*

$$\mathsf{W}_k^{(n)}(\mathcal{H}^{(n)}) \;:=\; \Big\{(e_1, \ldots, e_k) \in E^k \;:\; C_n\big(H(e_i), T(e_{i+1})\big) \text{ for } i = 1, \ldots, k-1\Big\}.$$

Definition 3.10 (Iterated line directed n-superhypergraphs)**.** Let $\mathcal{H}^{(n)}$ be as above. Define $L_n^1\big(\mathcal{H}^{(n)}\big) := L_n\big(\mathcal{H}^{(n)}\big)$ and, for $k \geq 1$,

$$L_n^{k+1}\big(\mathcal{H}^{(n)}\big) \;:=\; L_n\Big(L_n^k\big(\mathcal{H}^{(n)}\big)\Big).$$

We call $L_n^k\big(\mathcal{H}^{(n)}\big)$ the *k-fold (or iterated) line directed n-superhypergraph* of $\mathcal{H}^{(n)}$.

Example 3.11 (Iterated line directed superhypergraphs: programs composed of teams)**.** Let V_0 be a finite set of people in an organization. At level 1, form *teams* as subsets of people:

$$\text{teams } M_1 = \{a, b\}, \quad M_2 = \{c, d\}, \quad M_3 = \{b, c\} \;\subseteq\; \mathrm{POWS}^1(V_0).$$

At level 2, form *programs* as collections of teams (elements of $\mathrm{POWS}^2(V_0)$):

$$P_1 = \{M_1\}, \qquad P_2 = \{M_3\}, \qquad P_3 = \{M_2\}.$$

Consider a directed 2–superhypergraph $\mathcal{H}^{(2)} = (V_2, E)$ with vertex set $V_2 = \{P_1, P_2, P_3\}$ and superhyperedges

$$e_A : \big(\{P_1\},\ \{P_2\}\big), \qquad e_B : \big(\{P_2\},\ \{P_3\}\big), \qquad e_C : \big(\{P_3\},\ \{P_1\}\big).$$

Each $e_\bullet$ expresses a *program-level dependency*: the (level–2) head of one superhyperedge (a collection of teams) must be contained in the (level–2) tail of the next for the workflow to proceed.

The line directed 2–superhypergraph $L_2(\mathcal{H}^{(2)})$ has $\{e_A, e_B, e_C\}$ as vertices and singleton–tail superhyperedges

$$(\{e_A\}, \{e_B\}), \quad (\{e_B\}, \{e_C\}), \quad (\{e_C\}, \{e_A\}),$$

because, e.g., the head of e_A is $\{P_2\}$ and the tail of e_B is $\{P_2\}$ at the same level, so the composability condition is satisfied. Iterating once more, $L_2^2(\mathcal{H}^{(2)}) = L_2(L_2(\mathcal{H}^{(2)}))$ has as vertices the above three superhyperedges, and contains

$$\big(\ \{(\{e_A\}, \{e_B\})\}\ ,\ (\{e_B\}, \{e_C\})\ \big),$$

which encodes the two–step program chain $e_A \to e_B$ ("handoff from program P_1 to P_2, then from P_2 to P_3"). In this way, $L_2^k(\mathcal{H}^{(2)})$ summarizes length–k hierarchically consistent handoff sequences between program–level dependencies, preserving the nested (teams–within–programs) structure.

Theorem 3.12 (Well-definedness and walk correspondence)**.** *Let $\mathcal{H}^{(n)} = (V, E)$ be a directed n-superhypergraph.*

(A) *$L_n\big(\mathcal{H}^{(n)}\big)$ is a directed n-superhypergraph (in particular, $E' \subseteq \mathrm{POWS}^n(E) \times \mathrm{POWS}^n(E)$). Hence each iterate $L_n^k\big(\mathcal{H}^{(n)}\big)$ is a directed n-superhypergraph.*

(B) *For every $k \geq 1$ there is a canonical bijection*

$$\iota_k^{(n)} \;:\; \mathsf{W}_k^{(n)}\big(\mathcal{H}^{(n)}\big) \;\xrightarrow{\;\cong\;}\; V\big(L_n^k\big(\mathcal{H}^{(n)}\big)\big),$$

given recursively by

$$\iota_1^{(n)}(e_1) := e_1, \qquad \iota_{k+1}^{(n)}(e_1, \ldots, e_{k+1}) := \Big(\, \sigma_n\big(\iota_k^{(n)}(e_1, \ldots, e_k)\big),\ \sigma_n\big(\iota_k^{(n)}(e_2, \ldots, e_{k+1})\big) \,\Big).$$

Under this identification, there is an arc from $\iota_k^{(n)}(e_1, \ldots, e_k)$ to $\iota_k^{(n)}(e_2, \ldots, e_{k+1})$ in $L_n^k\big(\mathcal{H}^{(n)}\big)$ for each $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}^{(n)}\big(\mathcal{H}^{(n)}\big)$, i.e., adjacency is the one-step shift of walks.

Proof. (A) By the Definition, vertices of $L_n(\mathcal{H}^{(n)})$ are elements of E. For each ordered pair $(e_1, e_2) \in E \times E$ with $C_n\big(H(e_1), T(e_2)\big)$, the arc we add has tail $\sigma_n(e_1) \in \mathrm{POWS}^n(E)$ and head $\sigma_n(e_2) \in \mathrm{POWS}^n(E)$, so $E' \subseteq \mathrm{POWS}^n(E) \times \mathrm{POWS}^n(E)$ as required.

(B) For $k = 1$ this is the identity $E \cong V(L_n(\mathcal{H}^{(n)}))$. Assume the claim for k. Then $V\big(L_n^{k+1}(\mathcal{H}^{(n)})\big) = E\big(L_n^k(\mathcal{H}^{(n)})\big)$. By the Definition, an arc of $L_n^k(\mathcal{H}^{(n)})$ has the form $\big(\sigma_n(u), \sigma_n(v)\big)$ with $u, v \in V\big(L_n^k(\mathcal{H}^{(n)})\big)$ and $C_n\big(H_k(u), T_k(v)\big)$, where T_k, H_k are the tail/head maps in $L_n^k(\mathcal{H}^{(n)})$. By the induction hypothesis each u and v is *uniquely* represented as $u = \iota_k^{(n)}(e_1, \ldots, e_k)$ and $v = \iota_k^{(n)}(e_2, \ldots, e_{k+1})$ for a walk $(e_1, \ldots, e_{k+1}) \in \mathsf{W}_{k+1}^{(n)}$. A direct unwinding of the definitions shows that the compatibility condition for the line step is *equivalent* to $C_n\big(H(e_i), T(e_{i+1})\big)$ at the base level, i.e., exactly the walk condition. Hence every vertex of L_n^{k+1} is of the displayed recursive form, giving the bijection and the shift adjacency. □

Theorem 3.13 (Generalization ladder)**.** *Let $n \geq 0$, $k \geq 1$, and $\mathcal{H}^{(n)} = (V, E)$ be a directed n-superhypergraph.*

(i) ***Directed superhypergraph:*** *$L_n^k(\mathcal{H}^{(n)})$ is a directed n-superhypergraph for all k (closure under iteration), by Theorem 3.12 (A).*

(ii) ***Iterated line digraphs** (n=0)**:*** *If $n = 0$, then $\mathcal{H}^{(0)}$ is a digraph $D = (V, A)$ and $L_0^k(\mathcal{H}^{(0)})$ is naturally isomorphic to the classical k-iterated line digraph $L^k(D)$: vertices on both sides correspond bijectively to directed length-k walks $(v_0, \ldots, v_k)$, and adjacency is one-step shift.*

(iii) ***Iterated line directed hypergraphs** (n=1)**:*** *If $n = 1$, then $L_1^k(\mathcal{H}^{(1)})$ is (canonically) isomorphic to the iterated line directed hypergraph obtained by iterating the standard line operator that declares an arc $\{e_1\} \to e_2$ whenever $H(e_1) \cap T(e_2) \neq \emptyset$. (The two versions differ only by the singleton identification $e \leftrightarrow \{e\}$.)*

(iv) ***Line directed superhypergraph** (k=1)**:*** *For general $n \geq 0$, $L_n^1(\mathcal{H}^{(n)}) = L_n(\mathcal{H}^{(n)})$ is exactly the line directed n-superhypergraph of the Definition.*

Proof. (i) is Theorem 3.12 (A).

(ii) When $n = 0$ we have $E \subseteq V \times V$, and C_0 is equality of the shared vertex: (u, v) is compatible with (v, w). Vertices of L_0^k are in bijection with length-k walks $(v_0, \ldots, v_k)$; an arc exists from $(v_0, \ldots, v_k)$ to $(v_1, \ldots, v_{k+1})$ iff (v_i, v_{i+1}) consecutively match—precisely the rule for $L^k(D)$.

(iii) For $n = 1$, $T(e), H(e) \subseteq V$ and C_1 is nonempty intersection. By Definition, L_1 adds an arc from $\sigma_1(e_1) = \{e_1\}$ to $\sigma_1(e_2) = \{e_2\}$ exactly when $H(e_1) \cap T(e_2) \neq \emptyset$. Identifying $\{e\} \leftrightarrow e$ yields the usual line directed hypergraph. Iteration respects this identification, giving the stated isomorphism for all k.

(iv) is tautological from Definition 3.10. □

4 Total Directed HyperGraph

4.1 Total Directed Graph

A total directed graph connects every vertex pair in a directed graph, respecting possible edge orientations between them (cf. [69–73]).

Definition 4.1 (Total Directed Graph)**.** Let $D = (V, A)$ be a finite digraph. The *total directed graph* of D, denoted $T(D)$, is the digraph defined as follows:

$$V\big(T(D)\big) = V \cup A,$$
$$A\big(T(D)\big) = A_{VV} \cup A_{AA} \cup A_{VA} \cup A_{AV},$$

where

$$\begin{aligned}
A_{VV} &:= \{\, (u, v) \in V \times V \mid (u, v) \in A \,\}, && \text{(vertex–vertex adjacency as in } D\text{)}\\
A_{AA} &:= \{\, \big((x, u), (u, y)\big) \in A \times A \,\}, && \text{(consecutive arcs in } D\text{)}\\
A_{VA} &:= \{\, (u, (u, v)) \in V \times A \mid (u, v) \in A \,\}, && \text{(tail–to–arc incidence)}\\
A_{AV} &:= \{\, ((u, v), v) \in A \times V \mid (u, v) \in A \,\}. && \text{(arc–to–head incidence)}
\end{aligned}$$

Equivalently: the vertices of $T(D)$ correspond to the *elements* of D (vertices and arcs), and two vertices of $T(D)$ are adjacent precisely when the corresponding elements of D are adjacent (either as adjacent vertices, consecutive arcs, or incident vertex/arc in the appropriate direction).

Example 4.2 (Total Directed Graph: a two-step workflow)**.** Let the digraph be $D = (V, A)$ with

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

Then the total directed graph $T(D)$ has vertex set

$$V\big(T(D)\big) = V \cup A = \{A, B, C,\ (A, B),\ (B, C)\}.$$

Its arc set splits into the four canonical families:

$$\begin{aligned} A_{VV} &= \{(A, B),\ (B, C)\}, && \text{(original vertex–vertex arcs)} \\ A_{AA} &= \{\ \big((A, B), (B, C)\big)\ \}, && \text{(consecutive arcs)} \\ A_{VA} &= \{\ \big(A, (A, B)\big),\ \big(B, (B, C)\big)\ \}, && \text{(tail–to–arc incidence)} \\ A_{AV} &= \{\ \big((A, B), B\big),\ \big((B, C), C\big)\ \}. && \text{(arc–to–head incidence)} \end{aligned}$$

Thus $T(D)$ simultaneously records adjacency of vertices, adjacency of arcs, and both directions of vertex–arc incidence.

4.2 Total Directed HyperGraph

A total directed hypergraph connects every vertex subset in a directed hypergraph, preserving directed incidence between subsets.

Definition 4.3 (Total directed hypergraph)**.** Let $H = (V, E)$ be a directed hypergraph. Its *total directed hypergraph* $T(H)$ is the directed hypergraph defined by

$$V\big(T(H)\big) \ :=\ V\ \cup\ E, \qquad E\big(T(H)\big)\ :=\ E_{VV}\ \cup\ E_{EE}\ \cup\ E_{VE}\ \cup\ E_{EV},$$

where the four families of hyperarcs are

$$\begin{aligned} E_{VV} &:= \Big\{\ \big(\{u\}, \{v\}\big)\ :\ \exists\, e \in E \text{ with } u \in T(e),\ v \in H(e)\ \Big\}, && \text{(vertex–to–vertex adjacency via a hyperarc)}, \\ E_{EE} &:= \Big\{\ \big(\{e_1\}, \{e_2\}\big)\ :\ e_1, e_2 \in E,\ H(e_1) \cap T(e_2) \neq \emptyset\ \Big\}, && \text{(consecutive hyperarcs)}, \\ E_{VE} &:= \Big\{\ \big(\{u\}, \{e\}\big)\ :\ e \in E,\ u \in T(e)\ \Big\}, && \text{(tail–vertex to incident hyperarc)}, \\ E_{EV} &:= \Big\{\ \big(\{e\}, \{v\}\big)\ :\ e \in E,\ v \in H(e)\ \Big\}. && \text{(hyperarc to head–vertex)}. \end{aligned}$$

Thus every hyperarc of $T(H)$ has singleton tail and head, so $T(H)$ is a directed hypergraph whose hyperarcs encode vertex–vertex reachability through a hyperarc, hyperarc composition, and the two directions of vertex–hyperarc incidence.

Example 4.4 (Total Directed HyperGraph: a simple multi-input production)**.** Let $H = (V, E)$ be a directed hypergraph with

$$V = \{x, y, z, w\}, \qquad E = \{\, e_1, e_2 \,\},$$

where

$$e_1 :\ \big(T(e_1), H(e_1)\big) = \big(\{x, y\}, \{z\}\big), \qquad e_2 :\ \big(T(e_2), H(e_2)\big) = \big(\{z\}, \{w\}\big).$$

The total directed hypergraph $T(H)$ has vertex set

$$V\big(T(H)\big) = V \cup E = \{x, y, z, w,\ e_1, e_2\},$$

and hyperarcs

$$\begin{aligned} E_{VV} &= \{(\{x\}, \{z\}),\ (\{y\}, \{z\}),\ (\{z\}, \{w\})\}, \\ E_{EE} &= \{(\{e_1\}, \{e_2\})\} \quad \text{since } H(e_1) \cap T(e_2) = \{z\} \neq \emptyset, \\ E_{VE} &= \{(\{x\}, \{e_1\}),\ (\{y\}, \{e_1\}),\ (\{z\}, \{e_2\})\}, \\ E_{EV} &= \{(\{e_1\}, \{z\}),\ (\{e_2\}, \{w\})\}. \end{aligned}$$

Hence $T(H)$ captures (i) vertex-to-vertex reachability through one hyperarc, (ii) composability of hyperarcs, and (iii) both vertex–hyperarc incidences.

Theorem 4.5 (Well-definedness)**.** *For every directed hypergraph $H = (V, E)$, the structure $T(H)$ of Definition 4.3 is a directed hypergraph.*

Proof. By construction $V\big(T(H)\big) = V \cup E$ is a finite nonempty set. Each element of $E_{VV} \cup E_{EE} \cup E_{VE} \cup E_{EV}$ is an ordered pair $\big(\{x\}, \{y\}\big)$ of nonempty subsets of $V \cup E$, hence is a valid hyperarc. Therefore $T(H)$ meets the definition of a directed hypergraph. □

Theorem 4.6 (Generalization of the total directed graph)**.** *Let $D = (V, A)$ be a finite digraph and regard D as a directed hypergraph $H_D = (V, E)$ by identifying each arc $(u, v) \in A$ with the hyperarc $e_{(u,v)} = \big(\{u\}, \{v\}\big) \in E$. Then $T(H_D)$ is naturally isomorphic to the classical total directed graph $T(D)$:*

$$\Phi : \ V\big(T(H_D)\big) = V \cup E \ \longrightarrow \ V\big(T(D)\big) = V \cup A, \quad \Phi(v) = v \ (v \in V), \ \ \Phi\big(e_{(u,v)}\big) = (u, v),$$

and $\big(\{x\}, \{y\}\big) \in E\big(T(H_D)\big)$ if and only if $\big(\Phi(x), \Phi(y)\big) \in A\big(T(D)\big)$.

Proof. The map Φ is a bijection by construction. We check the four arc families.

(i) Vertex–vertex: $(\{u\}, \{v\}) \in E_{VV}$ for $T(H_D)$ iff there exists $e \in E$ with $u \in T(e)$ and $v \in H(e)$. Since every $e \in E$ has the form $e_{(x,y)} = \big(\{x\}, \{y\}\big)$, this is equivalent to $(u, v) \in A$. Thus $\big(\Phi(u), \Phi(v)\big) = (u, v) \in A_{VV}$ of $T(D)$.

(ii) Hyperarc–hyperarc: $\big(\{e_{(x,u)}\}, \{e_{(u,y)}\}\big) \in E_{EE}$ for $T(H_D)$ iff $H\big(e_{(x,u)}\big) \cap T\big(e_{(u,y)}\big) = \{u\} \cap \{u\} \neq \emptyset$, i.e., the head of the first equals the tail of the second. Under Φ this becomes $\big((x, u), (u, y)\big) \in A_{AA}$ of $T(D)$.

(iii) Vertex–hyperarc incidence: $\big(\{u\}, \{e_{(u,v)}\}\big) \in E_{VE}$ for $T(H_D)$ iff $u \in T\big(e_{(u,v)}\big)$, which corresponds to $\big(u, (u, v)\big) \in A_{VA}$ of $T(D)$.

(iv) Hyperarc–vertex incidence: $\big(\{e_{(u,v)}\}, \{v\}\big) \in E_{EV}$ for $T(H_D)$ iff $v \in H\big(e_{(u,v)}\big)$, which corresponds to $\big((u, v), v\big) \in A_{AV}$ of $T(D)$.

Thus Φ preserves and reflects all arcs, yielding an isomorphism $T(H_D) \cong T(D)$. □

4.3 Total Directed SuperHyperGraph

A total directed superhypergraph connects all vertex collections in a directed superhypergraph, maintaining directed incidence across hierarchical levels. We will need two canonical constructions: the n-fold *singleton lift* into $\mathrm{POWS}^n(X)$, and the *atom map* that flattens an n-level object over V to the set of base vertices it contains.

Definition 4.7 (Singleton lift and atom map)**.** For a set X and $n \geq 0$, define the n-fold singleton lift $\sigma_n : X \longrightarrow \mathrm{POWS}^n(X)$ recursively by

$$\sigma_0(x) = x, \qquad \sigma_{k+1}(x) = \{\sigma_k(x)\} \quad (k \geq 0).$$

For a finite base set V define the atom map $\mathrm{At}_n : \mathrm{POWS}^n(V) \longrightarrow \mathrm{POWS}(V)$ recursively by

$$\mathrm{At}_0(v) = \{v\}, \qquad \mathrm{At}_{k+1}(X) = \bigcup_{Y \in X} \mathrm{At}_k(Y) \quad (k \geq 0).$$

Thus At_n collects all level-0 (i.e. base) vertices appearing inside a nested object in $\mathrm{POWS}^n(V)$.

Definition 4.8 (Total directed n-superhypergraph)**.** Let $S = (V, E)$ be a directed n-superhypergraph as in the Definition. Form its *total directed n-superhypergraph* $T^{(n)}(S)$ as follows.

Vertex set. Let $\widetilde{V} := V \ \cup \ E$, i.e., treat every original edge $e \in E$ as a new atomic vertex. Set $V\big(T^{(n)}(S)\big) := \widetilde{V}$.

Edge set. Define four families of directed n-superhyperedges, each given by an ordered pair in $\mathrm{POWS}^n(\widetilde{V}) \times \mathrm{POWS}^n(\widetilde{V})$ using the n-fold singleton lift σ_n:

$$
\begin{aligned}
E_{VV}^{(n)} &:= \Big\{ \big(\sigma_n(u),\ \sigma_n(v)\big)\ :\ \exists\, e \in E \text{ with } u \in \mathrm{At}_n\big(T(e)\big),\ v \in \mathrm{At}_n\big(H(e)\big) \Big\},\\
E_{EE}^{(n)} &:= \Big\{ \big(\sigma_n(e_1),\ \sigma_n(e_2)\big)\ :\ e_1, e_2 \in E,\ \mathrm{At}_n\big(H(e_1)\big) \cap \mathrm{At}_n\big(T(e_2)\big) \neq \emptyset \Big\},\\
E_{VE}^{(n)} &:= \Big\{ \big(\sigma_n(u),\ \sigma_n(e)\big)\ :\ e \in E,\ u \in \mathrm{At}_n\big(T(e)\big) \Big\},\\
E_{EV}^{(n)} &:= \Big\{ \big(\sigma_n(e),\ \sigma_n(v)\big)\ :\ e \in E,\ v \in \mathrm{At}_n\big(H(e)\big) \Big\}.
\end{aligned}
$$

The edge set of $T^{(n)}(S)$ is the union

$$
E\big(T^{(n)}(S)\big)\ :=\ E_{VV}^{(n)}\ \cup\ E_{EE}^{(n)}\ \cup\ E_{VE}^{(n)}\ \cup\ E_{EV}^{(n)}\ \subseteq\ \mathrm{POWS}^n(\widetilde{V}) \times \mathrm{POWS}^n(\widetilde{V}).
$$

Example 4.9 (Total Directed SuperHyperGraph (level $n = 2$): programs built from teams)**.** Let the base set be $V = \{a, b, c, d\}$. Consider a directed 2–superhypergraph $S = (V, E)$ with two 2–superhyperedges

$$
e_A :\ \big(T(e_A), H(e_A)\big) = \big(\{\{a, b\}\},\ \{\{c\}\}\big), \qquad e_B :\ \big(T(e_B), H(e_B)\big) = \big(\{\{c\}\},\ \{\{d\}\}\big),
$$

where elements of $\mathrm{POWS}^2(V)$ are sets of subsets of V. Write $\sigma_2(x) = \{\{x\}\}$ for the double singleton lift and At_2 for the atom map (which flattens to base vertices), so $\mathrm{At}_2(T(e_A)) = \{a, b\}$, $\mathrm{At}_2(H(e_A)) = \{c\}$, $\mathrm{At}_2(T(e_B)) = \{c\}$, $\mathrm{At}_2(H(e_B)) = \{d\}$. The total directed 2–superhypergraph $T^{(2)}(S)$ has vertex set

$$
\widetilde{V} = V \cup E = \{a, b, c, d,\ e_A, e_B\}.
$$

Its 2–level edges are the following four families (all tails/heads live in $\mathrm{POWS}^2(\widetilde{V})$ via σ_2):

$$
\begin{aligned}
E_{VV}^{(2)} &= \big\{ (\sigma_2(a), \sigma_2(c)),\ (\sigma_2(b), \sigma_2(c)),\ (\sigma_2(c), \sigma_2(d)) \big\},\\
E_{EE}^{(2)} &= \big\{ (\sigma_2(e_A), \sigma_2(e_B)) \big\} \quad \text{since } \mathrm{At}_2\big(H(e_A)\big) \cap \mathrm{At}_2\big(T(e_B)\big) = \{c\},\\
E_{VE}^{(2)} &= \big\{ (\sigma_2(a), \sigma_2(e_A)),\ (\sigma_2(b), \sigma_2(e_A)),\ (\sigma_2(c), \sigma_2(e_B)) \big\},\\
E_{EV}^{(2)} &= \big\{ (\sigma_2(e_A), \sigma_2(c)),\ (\sigma_2(e_B), \sigma_2(d)) \big\}.
\end{aligned}
$$

Thus $T^{(2)}(S)$ elevates both base vertices and 2–level edges to vertices and encodes vertex–vertex reachability through a single 2–edge, composition of 2–edges, and both directions of incidence, all placed coherently at level 2.

Intuitively, $T^{(n)}(S)$ makes both *objects* of S (the base vertices V and the n-level edges E) into vertices, and then creates four types of directed n-superhyperedges encoding: (i) vertex-to-vertex reachability through a single original edge; (ii) composition of original edges; (iii) incidence from tail-vertices to an edge; and (iv) incidence from an edge to its head-vertices. The use of At_n ensures these are decided at the level of base vertices even when $T(e), H(e)$ are nested ($n \geq 1$). The lift σ_n places these incidences at the correct superlevel.

Theorem 4.10 (Well-definedness: $T^{(n)}(S)$ is a directed n-superhypergraph)**.** *For any directed n-superhypergraph $S = (V, E)$, the structure $T^{(n)}(S) = \big(\widetilde{V},\ E\big(T^{(n)}(S)\big)\big)$ of Definition 4.8 is a directed n-superhypergraph on the base set $\widetilde{V}$.*

Proof. By construction $\widetilde{V} = V \cup E$ is finite and nonempty. Each edge of $T^{(n)}(S)$ has the form $\big(\sigma_n(x), \sigma_n(y)\big)$ with $x, y \in \widetilde{V}$; hence its tail and head lie in $\mathrm{POWS}^n(\widetilde{V})$ by the definition of σ_n. Therefore $E\big(T^{(n)}(S)\big) \subseteq \mathrm{POWS}^n(\widetilde{V}) \times \mathrm{POWS}^n(\widetilde{V})$, as required in the Definition. □

Theorem 4.11 (Generalization of the total directed graph)**.** *Let $D = (V, A)$ be a finite digraph and regard it as a directed 0-superhypergraph $S_0 = (V, E_0)$ with $E_0 = A \subseteq V \times V$. Then the total directed 0-superhypergraph $T^{(0)}(S_0)$ is (canonically) isomorphic to the classical total directed graph $T(D)$.*

Proof. Here $n = 0$, σ_0 is the identity, and $\mathrm{At}_0(v) = \{v\}$. The vertex set of $T^{(0)}(S_0)$ is $\widetilde{V} = V \cup E_0 = V \cup A$, which equals the vertex set of $T(D)$. We verify the four edge families coincide:

(i) $E^{(0)}_{VV}$ consists of (u, v) whenever $\exists\,(x, y) \in A$ with $u \in \{x\}$ and $v \in \{y\}$, i.e. precisely when $(u, v) \in A$. This is A_{VV} in the definition of $T(D)$.

(ii) $E^{(0)}_{EE}$ contains $\big((x, u), (u, y)\big)$ whenever $(x, u), (u, y) \in A$, i.e. consecutive arcs; this equals A_{AA} in $T(D)$.

(iii) $E^{(0)}_{VE}$ contains $\big(u, (u, v)\big)$ for each $(u, v) \in A$ (tail incidence), i.e. A_{VA}.

(iv) $E^{(0)}_{EV}$ contains $\big((u, v), v\big)$ for each $(u, v) \in A$ (head incidence), i.e. A_{AV}.

Thus $T^{(0)}(S_0) = T(D)$. □

Theorem 4.12 (Generalization of the total directed hypergraph)**.** *Let $H = (V, E)$ be a directed hypergraph (i.e. a directed 1-superhypergraph). Then the total directed 1-superhypergraph $T^{(1)}(H)$ is naturally isomorphic to the total directed hypergraph $T(H)$ of Definition* (Total directed hypergraph).

Proof. Now $n = 1$, so $\sigma_1(x) = \{x\}$ and At_1 reduces a subset of V to itself. The vertex set of $T^{(1)}(H)$ is $\widetilde{V} = V \cup E$, which is the vertex set used in $T(H)$. For edges, note that every edge in $T^{(1)}(H)$ has singleton tail and head, so it corresponds to an ordered pair of *atoms* in $V \cup E$.

(i) $E^{(1)}_{VV}$ contains $\big(\{u\}, \{v\}\big)$ iff there exists $e \in E$ with $u \in T(e)$ and $v \in H(e)$, i.e. exactly the vertex–vertex arcs in $T(H)$. Identifying $\{u\} \leftrightarrow u$ and $\{v\} \leftrightarrow v$ yields the same arcs as in E_{VV} of $T(H)$.

(ii) $E^{(1)}_{EE}$ contains $\big(\{e_1\}, \{e_2\}\big)$ iff $H(e_1) \cap T(e_2) \neq \emptyset$, precisely the consecutive-hyperarc condition in $T(H)$. Under $\{e\} \leftrightarrow e$ this is E_{EE} of $T(H)$.

(iii) $E^{(1)}_{VE}$ contains $\big(\{u\}, \{e\}\big)$ iff $u \in T(e)$, matching E_{VE} of $T(H)$.

(iv) $E^{(1)}_{EV}$ contains $\big(\{e\}, \{v\}\big)$ iff $v \in H(e)$, matching E_{EV} of $T(H)$.

Hence the identification of singleton lifts gives a canonical isomorphism $T^{(1)}(H) \cong T(H)$. □

5 Line Bidirected Graph and Total Bidirected Graph

5.1 Bidirected Graph

A bidirected graph is a graph where each edge endpoint has an independent orientation, allowing bidirectional incidence representation [74–76].

Definition 5.1 (bidirected graph)**.** [75, 77, 78] A *bidirected graph* is a directed graph with independent orientations at each endpoint of an edge. Formally, a bidirected graph is defined as:

$$G = (V, E, \tau),$$

where:

- V is the set of vertices.
- E is the set of edges, each connecting two vertices.
- $\tau : V \times E \to \{-1, 0, 1\}$ is the bidirection function:
  - $\tau(v, e) = 1$: Edge e is directed toward vertex v.

- $\tau(v, e) = -1$: Edge e is directed away from vertex v.
- $\tau(v, e) = 0$: Vertex v is not incident to edge e.

Example 5.2 (A simple Bidirected Graph)**.** Let $V = \{A, B, C\}$ and let $E = \{e_1, e_2, e_3\}$ with the underlying (undirected) incidences

$$e_1 \text{ joins } A \text{ and } B, \qquad e_2 \text{ joins } B \text{ and } C, \qquad e_3 \text{ joins } C \text{ and } A.$$

Define the bidirection function $\tau : V \times E \to \{-1, 0, +1\}$ by

$$\tau(A, e_1) = -1,\ \tau(B, e_1) = +1; \qquad \tau(B, e_2) = -1,\ \tau(C, e_2) = +1; \qquad \tau(C, e_3) = -1,\ \tau(A, e_3) = +1,$$

and $\tau(v, e) = 0$ if v is not incident with e. Intuitively, each edge is oriented "from -1 to $+1$", so this encodes the directed 3-cycle $A \to B \to C \to A$ while keeping endpoint orientations independent (the essence of bidirected graphs).

5.2 Line Bidirected Graph

A line bidirected graph represents edges of a bidirected graph as vertices, connecting them if they share an endpoint in the original.

For $v \in V$ write

$$E^+(v) := \{ e \in E : \tau(v, e) = +1 \}, \qquad E^-(v) := \{ e \in E : \tau(v, e) = -1 \},$$

the incident edges that *enter* (resp. *leave*) v.

Definition 5.3 (Line bidirected graph)**.** Let $G = (V, E, \tau)$ be a finite bidirected graph. Its *line bidirected graph* is the bidirected graph

$$L_{\mathrm{bi}}(G) = \big(E,\ E_L,\ \tau_L\big),$$

defined as follows.

- **Vertices:** $V\big(L_{\mathrm{bi}}(G)\big) = E$; that is, each edge of G becomes a vertex of $L_{\mathrm{bi}}(G)$.
- **Edges:** For each $v \in V$ and each *ordered* pair $(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E^+(v) \times E^-(v)$ with $e_{\mathrm{in}} \neq e_{\mathrm{out}}$, introduce a new edge $\ell = \ell_v(e_{\mathrm{in}}, e_{\mathrm{out}})$ of $L_{\mathrm{bi}}(G)$ incident with the two vertices e_{in} and e_{out} and with endpoint signs

  $$\tau_L\big(e_{\mathrm{in}}, \ell\big) = -1, \qquad \tau_L\big(e_{\mathrm{out}}, \ell\big) = +1, \qquad \tau_L\big(e, \ell\big) = 0 \ \text{ for } e \notin \{e_{\mathrm{in}}, e_{\mathrm{out}}\}.$$

  (If desired, parallel edges created by different v or by different ordered pairs may be kept, yielding a bidirected multigraph; otherwise identify parallel copies.)

Intuitively, each $\ell_v(e_{\mathrm{in}}, e_{\mathrm{out}})$ encodes a *feasible turn at* v from the incoming end of e_{in} to the outgoing end of e_{out}.

Example 5.4 (Line Bidirected Graph of the above instance)**.** For $v \in V$, set

$$E^+(v) := \{ e \in E : \tau(v, e) = +1 \}, \qquad E^-(v) := \{ e \in E : \tau(v, e) = -1 \}.$$

With the τ from the previous example we have

$$E^+(A) = \{e_3\},\ E^-(A) = \{e_1\}; \quad E^+(B) = \{e_1\},\ E^-(B) = \{e_2\}; \quad E^+(C) = \{e_2\},\ E^-(C) = \{e_3\}.$$

The *line bidirected graph* $L_{\mathrm{bi}}(G) = (E, E_L, \tau_L)$ has vertex set E and, for each $v \in V$ and ordered pair $(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E^+(v) \times E^-(v)$ with $e_{\mathrm{in}} \neq e_{\mathrm{out}}$, an edge $\ell_v(e_{\mathrm{in}}, e_{\mathrm{out}})$ such that

$$\tau_L(e_{\mathrm{in}}, \ell_v) = -1, \qquad \tau_L(e_{\mathrm{out}}, \ell_v) = +1, \qquad \tau_L(e, \ell_v) = 0 \text{ for } e \notin \{e_{\mathrm{in}}, e_{\mathrm{out}}\}.$$

Thus here

$$E_L = \big\{\ \ell_A(e_3, e_1),\ \ell_B(e_1, e_2),\ \ell_C(e_2, e_3)\ \big\},$$

with signed incidences

$$\tau_L(e_3, \ell_A) = -1,\ \tau_L(e_1, \ell_A) = +1;$$
$$\tau_L(e_1, \ell_B) = -1,\ \tau_L(e_2, \ell_B) = +1;$$
$$\tau_L(e_2, \ell_C) = -1,\ \tau_L(e_3, \ell_C) = +1.$$

If we orient each line edge from its -1 endpoint to its $+1$ endpoint, we obtain the directed 3-cycle $e_3 \to e_1 \to e_2 \to e_3$, which coincides with the classical line digraph of $A \to B \to C \to A$.

Theorem 5.5 (Well-definedness)**.** *For every finite bidirected graph $G = (V, E, \tau)$, the structure $\mathbb{L}_{\mathrm{bi}}(G) = \big(E, E_L, \tau_L\big)$ in Definition 5.3 is a (finite) bidirected graph.*

Proof. By construction the vertex set is E (finite). Each edge $\ell = \ell_v(e_{\mathrm{in}}, e_{\mathrm{out}})$ is incident with exactly two vertices e_{in} and e_{out} and has $\tau_L(e_{\mathrm{in}}, \ell) = -1$, $\tau_L(e_{\mathrm{out}}, \ell) = +1$ and zero elsewhere. Thus $\tau_L : E \times E_L \to \{-1, 0, +1\}$ is a valid bidirected incidence function, and $\mathbb{L}_{\mathrm{bi}}(G)$ is bidirected by Definition. □

Theorem 5.6 (Generalizes the line digraph)**.** *Let $D = (V, A)$ be a finite digraph and view it as a bidirected graph $G = (V, E, \tau)$ by taking $E = A$ and, for each arc $e = (u, v)$,*

$$\tau(u, e) = -1 \quad (\textit{tail}), \qquad \tau(v, e) = +1 \quad (\textit{head}).$$

Orient each edge ℓ of $\mathbb{L}_{\mathrm{bi}}(G)$ from *its -1 endpoint* to *its $+1$ endpoint. Then the resulting digraph is canonically isomorphic to the classical line digraph $L(D)$.*

Proof. Fix arcs $e_1 = (x, v)$ and $e_2 = (v, y)$ of D with $\mathrm{head}(e_1) = v = \mathrm{tail}(e_2)$. In the bidirected encoding, $\tau(v, e_1) = +1$ and $\tau(v, e_2) = -1$, so by Definition 5.3 there is an edge $\ell_v(e_1, e_2)$ with $\tau_L(e_1, \ell) = -1$ and $\tau_L(e_2, \ell) = +1$. Under the "$- \to +$" orientation rule, this yields an arc $e_1 \to e_2$. Conversely, any edge ℓ of $\mathbb{L}_{\mathrm{bi}}(G)$ arises from some v and ordered pair (e_1, e_2) with $\tau(v, e_1) = +1$, $\tau(v, e_2) = -1$, which in the digraph corresponds exactly to $\mathrm{head}(e_1) = v = \mathrm{tail}(e_2)$, i.e. a directed adjacency $e_1 \to e_2$ in $L(D)$. This gives a bijection between arcs of $L(D)$ and oriented edges of $\mathbb{L}_{\mathrm{bi}}(G)$, preserving incidence, hence a canonical isomorphism of digraphs. □

5.3 Total Bidirected Graph

A total bidirected graph extends a bidirected graph by adding edges between all vertex pairs with any endpoint orientation relationship present.

Definition 5.7 (Total bidirected graph)**.** Let $G = (V, E, \tau)$ be a finite bidirected graph. The *total bidirected graph* of G is the bidirected graph

$$\mathbb{T}_{\mathrm{bi}}(G) \;=\; \big(V_T, E_T, \tau_T\big),$$

defined as follows.

Vertices. $V_T := V \cup E$; thus both original vertices and original edges of G are vertices of $\mathbb{T}_{\mathrm{bi}}(G)$.

Edges. $E_T := E_{VV} \;\cup\; E_{EE} \;\cup\; E_{VE}$, where

$$\begin{aligned} E_{VV} &:= \big\{ \kappa_e \;\big|\; e \in E \text{ with endpoints } u, v \in V \text{ (possibly } u = v\text{)} \big\}, \\ E_{EE} &:= \big\{ \lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}}) \;\big|\; v \in V,\ e_{\mathrm{in}} \in E^+(v),\ e_{\mathrm{out}} \in E^-(v),\ e_{\mathrm{in}} \neq e_{\mathrm{out}} \big\}, \\ E_{VE} &:= \big\{ \iota(v, e) \;\big|\; v \in V,\ e \in E,\ \tau(v, e) \in \{\pm 1\} \big\}. \end{aligned}$$

Incidence signs. The signed incidences $\tau_T(\cdot, \cdot)$ are given by

(V–V edges) if e has endpoints u, v in G, then $\kappa_e \in E_{VV}$ with
$\tau_T(u, \kappa_e) = \tau(u, e), \qquad \tau_T(v, \kappa_e) = \tau(v, e), \qquad \tau_T(x, \kappa_e) = 0 \ \ \forall x \in V_T \setminus \{u, v\};$

(E–E edges) $\lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E_{EE}$ is incident with $e_{\mathrm{in}}, e_{\mathrm{out}}$ via
$\tau_T\big(e_{\mathrm{in}}, \lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}})\big) = -1, \qquad \tau_T\big(e_{\mathrm{out}}, \lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}})\big) = +1,$
$\tau_T(x, \lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}})) = 0 \ \ \forall x \notin \{e_{\mathrm{in}}, e_{\mathrm{out}}\};$

(V–E edges) $\iota(v, e) \in E_{VE}$ is incident with v and e via
$\tau_T\big(v, \iota(v, e)\big) = \tau(v, e), \qquad \tau_T\big(e, \iota(v, e)\big) = -\,\tau(v, e), \qquad \tau_T(x, \iota(v, e)) = 0 \ \ \forall x \notin \{v, e\}.$

(If desired, parallel edges created by distinct witnesses may be kept, producing a bidirected multigraph.)

Example 5.8 (Total Bidirected Graph of the same instance)**.** Let $G = (V, E, \tau)$ be the bidirected graph in the first example. The *total bidirected graph* $T_{\mathrm{bi}}(G) = (V_T, E_T, \tau_T)$ has

$$V_T = V \cup E = \{A, B, C, e_1, e_2, e_3\},$$

and three edge families: vertex–vertex (E_{VV}), edge–edge (E_{EE}), and vertex–edge (E_{VE}).

(i) Vertex–vertex edges. For each $e \in E$ with endpoints u, v in G, add κ_e incident to u, v with

$$\tau_T(u, \kappa_e) = \tau(u, e), \qquad \tau_T(v, \kappa_e) = \tau(v, e).$$

Here:

$$\tau_T(A, \kappa_{e_1}) = -1,\ \tau_T(B, \kappa_{e_1}) = +1; \quad \tau_T(B, \kappa_{e_2}) = -1,\ \tau_T(C, \kappa_{e_2}) = +1; \quad \tau_T(C, \kappa_{e_3}) = -1,\ \tau_T(A, \kappa_{e_3}) = +1.$$

(ii) Edge–edge edges. For each $v \in V$ and $(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E^+(v) \times E^-(v)$, add $\lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}})$ incident with $e_{\mathrm{in}}, e_{\mathrm{out}}$ by

$$\begin{aligned} \tau_T\big(e_{\mathrm{in}}, \lambda_v\big) &= -1, \\ \tau_T\big(e_{\mathrm{out}}, \lambda_v\big) &= +1. \end{aligned}$$

Thus:

$$\begin{aligned} \tau_T(e_1, \lambda_B(e_1, e_2)) = -1,\ \tau_T(e_2, \lambda_B(e_1, e_2)) = +1; \\ \tau_T(e_2, \lambda_C(e_2, e_3)) = -1,\ \tau_T(e_3, \lambda_C(e_2, e_3)) = +1; \\ \tau_T(e_3, \lambda_A(e_3, e_1)) = -1,\ \tau_T(e_1, \lambda_A(e_3, e_1)) = +1. \end{aligned}$$

(iii) Vertex–edge incidence edges. For each incident pair (v, e) with $\tau(v, e) \in \{\pm 1\}$, add $\iota(v, e)$ with

$$\begin{aligned} \tau_T\big(v, \iota(v, e)\big) &= \tau(v, e), \\ \tau_T\big(e, \iota(v, e)\big) &= -\tau(v, e). \end{aligned}$$

Concretely:

$$\begin{array}{ll} \tau_T\big(A, \iota(A, e_1)\big) = -1,\ \tau_T\big(e_1, \iota(A, e_1)\big) = +1; & \tau_T\big(B, \iota(B, e_1)\big) = +1,\ \tau_T\big(e_1, \iota(B, e_1)\big) = -1; \\ \tau_T\big(B, \iota(B, e_2)\big) = -1,\ \tau_T\big(e_2, \iota(B, e_2)\big) = +1; & \tau_T\big(C, \iota(C, e_2)\big) = +1,\ \tau_T\big(e_2, \iota(C, e_2)\big) = -1; \\ \tau_T\big(C, \iota(C, e_3)\big) = -1,\ \tau_T\big(e_3, \iota(C, e_3)\big) = +1; & \tau_T\big(A, \iota(A, e_3)\big) = +1,\ \tau_T\big(e_3, \iota(A, e_3)\big) = -1. \end{array}$$

Hence $T_{\mathrm{bi}}(G)$ elevates both original vertices and edges to vertices and connects them by (a) vertex–vertex edges respecting the original endpoint signs, (b) edge–edge edges encoding feasible turns at intermediate vertices, and (c) vertex–edge edges encoding signed incidence.

Theorem 5.9 (Well-definedness)**.** *For every finite bidirected graph $G = (V, E, \tau)$, the structure $T_{\mathrm{bi}}(G) = (V_T, E_T, \tau_T)$ of Definition 5.7 is a finite bidirected graph.*

Proof. By construction, V_T is finite. For each edge of E_T there are exactly two nonzero incidences, each in $\{\pm 1\}$, and all other incidences are 0:

- For $\kappa_e \in E_{VV}$, only the endpoints u, v of e have nonzero incidence, with $\tau_T(u, \kappa_e) = \tau(u, e)$ and $\tau_T(v, \kappa_e) = \tau(v, e)$.
- For $\lambda_v(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E_{EE}$, only e_{in} and e_{out} have nonzero incidence, with values -1 and $+1$ respectively.
- For $\iota(v, e) \in E_{VE}$, only v and e have nonzero incidence, with $\tau_T(v, \iota(v, e)) = \tau(v, e)$ and $\tau_T(e, \iota(v, e)) = -\tau(v, e)$.

Thus $\tau_T : V_T \times E_T \to \{-1, 0, +1\}$ satisfies the bidirected axioms of Definition. □

Theorem 5.10 (Generalizes the total directed graph)**.** *Let $D = (V, A)$ be a finite digraph and view it as a bidirected graph $G = (V, E, \tau)$ by taking $E = A$ and, for each arc $e = (u, v)$,*

$$\tau(u, e) = -1 \quad (\textit{tail}), \qquad \tau(v, e) = +1 \quad (\textit{head}).$$

Orient each edge of $T_{\mathrm{bi}}(G)$ from its -1 endpoint to its $+1$ endpoint. The resulting digraph is canonically isomorphic to the classical total directed graph $T(D)$ of Definition (Total Directed Graph).

Proof. We show a bijection between the four arc classes of $T(D)$ and the oriented edges of $T_{\mathrm{bi}}(G)$.

(VV-class). For each arc $e = (u, v) \in A$, the edge $\kappa_e \in E_{VV}$ satisfies $\tau_T(u, \kappa_e) = -1$ and $\tau_T(v, \kappa_e) = +1$, hence orients as $u \to v$. This yields exactly the arc set $A_{VV} = \{(u, v) \in V \times V : (u, v) \in A\}$.

(EE-class). For consecutive arcs $e_1 = (x, u)$ and $e_2 = (u, y)$ in D ($\mathrm{head}(e_1) = u = \mathrm{tail}(e_2)$), we have $\tau(u, e_1) = +1$ and $\tau(u, e_2) = -1$, so $\lambda_u(e_1, e_2) \in E_{EE}$ satisfies $\tau_T(e_1, \lambda) = -1$, $\tau_T(e_2, \lambda) = +1$, hence orients as $e_1 \to e_2$. This yields exactly $A_{AA} = \{((x, u), (u, y)) \in A \times A\}$.

(VA-class). If $e = (u, v) \in A$, then $\tau(u, e) = -1$, so $\iota(u, e) \in E_{VE}$ satisfies $\tau_T(u, \iota(u, e)) = -1$, $\tau_T(e, \iota(u, e)) = +1$, hence orients as $u \to e$. This yields exactly $A_{VA} = \{(u, (u, v)) : (u, v) \in A\}$.

(AV-class). If $e = (u, v) \in A$, then $\tau(v, e) = +1$, so $\iota(v, e) \in E_{VE}$ satisfies $\tau_T(v, \iota(v, e)) = +1$, $\tau_T(e, \iota(v, e)) = -1$, hence orients as $e \to v$. This yields exactly $A_{AV} = \{((u, v), v) : (u, v) \in A\}$.

These four correspondences are bijective on their respective classes and preserve incidence. Therefore the orientation of $T_{\mathrm{bi}}(G)$ from -1 to $+1$ produces a digraph canonically isomorphic to $T(D)$. □

6 Line Multidirected Graph

6.1 Multidirected Graph

A multidirected graph is a graph that allows multiple directed edges between vertices, each assigned a specific multiplicity, thereby extending the standard directed graph representation [79–81].

Definition 6.1 (Multidirected Graph)**.** [79, 82] A *multidirected graph* is a tuple

$$G = (V, E, s, t, m),$$

where

- V is a finite set of vertices,
- E is a finite set of edges,
- $s, t : E \to V$ assign to each edge its source and target,
- $m : V \times V \to \mathbb{N}_0$ gives the multiplicity of edges from one vertex to another.

Example 6.2 (Multidirected Graph: parallel shipments in a small logistics hub)**.** Let $V = \{A, B, C\}$ denote depots. Consider four directed shipments (edges)

$$E = \{e_1, e_2, e_3, e_4\}, \qquad s(e_1) = A,\ t(e_1) = B;\ \ s(e_2) = A,\ t(e_2) = B;$$

$$s(e_3) = B,\ t(e_3) = C;\ \ s(e_4) = B,\ t(e_4) = B \text{ (a loop at } B).$$

The multiplicity function $m : V \times V \to \mathbb{N}_0$ records how many parallel edges exist from one depot to another:

$$m(A, B) = 2, \quad m(B, C) = 1, \quad m(B, B) = 1,$$

and $m(u, v) = 0$ for all remaining ordered pairs (u, v). Then $G = (V, E, s, t, m)$ is a multidirected graph: there are two parallel edges $A \to B$ (namely e_1, e_2), one edge $B \to C$ (e_3), and one loop at B (e_4).

6.2 Line Multidirected Graph

A line multidirected graph is constructed from a multidirected graph by taking its vertices to represent the edges of the original, with edges indicating consecutive incidence in the original graph.

Definition 6.3 (Line Multidirected Graph)**.** Let $G = (V, E, s, t, m)$ be a finite multidirected graph. The *line multidirected graph* of G, denoted $L_{\mathrm{md}}(G)$, is the multidirected graph

$$L_{\mathrm{md}}(G) \;=\; \big(V_L, E_L, s_L, t_L, m_L\big)$$

defined as follows.

- **Vertices.** $V_L := E$. Thus each edge of G becomes a vertex of $L_{\mathrm{md}}(G)$.
- **Edges.** Set
  $$E_L \;:=\; \{\, \epsilon = (e_1, e_2) \in E \times E \;:\; t(e_1) = s(e_2) \,\}.$$
  Each ordered pair of *consecutive* edges in G is an edge of $L_{\mathrm{md}}(G)$.
- **Source/Target.** For $\epsilon = (e_1, e_2) \in E_L$ put
  $$s_L(\epsilon) := e_1, \qquad t_L(\epsilon) := e_2.$$
- **Multiplicity.** For $x, y \in V_L = E$, define
  $$m_L(x, y) \;:=\; \#\big\{\, \epsilon \in E_L \;:\; s_L(\epsilon) = x,\; t_L(\epsilon) = y \,\big\}.$$
  Equivalently, $m_L(e_1, e_2) = 1$ if $t(e_1) = s(e_2)$ and 0 otherwise (so multiplicities count parallel line-edges should a modeling choice create them).

Loops in G are permitted and induce the natural loops in $L_{\mathrm{md}}(G)$ when $t(e) = s(e)$.

Example 6.4 (Line Multidirected Graph of the logistics hub)**.** Let $G = (V, E, s, t, m)$ be as above. Its line multidirected graph $L_{\mathrm{md}}(G) = (V_L, E_L, s_L, t_L, m_L)$ has

$$V_L = E = \{e_1, e_2, e_3, e_4\}.$$

By definition,

$$E_L = \{\, \epsilon = (e_i, e_j) \in E \times E \;:\; t(e_i) = s(e_j) \,\}.$$

Since $t(e_1) = t(e_2) = B$, and the edges starting at B are e_3 and e_4, while $t(e_3) = C$ (no outgoing edge from C in this instance) and $t(e_4) = B$ (with outgoing e_3, e_4 again), we obtain

$$E_L = \big\{(e_1, e_3),\ (e_1, e_4),\ (e_2, e_3),\ (e_2, e_4),\ (e_4, e_3),\ (e_4, e_4)\big\}.$$

For each $\epsilon = (e_i, e_j) \in E_L$ we set $s_L(\epsilon) = e_i$ and $t_L(\epsilon) = e_j$. The multiplicity on $V_L \times V_L$ counts how many such ordered pairs realize the same ordered pair of vertices:

$$m_L(e_i, e_j) = \#\{\epsilon \in E_L :\ s_L(\epsilon) = e_i,\ t_L(\epsilon) = e_j\}.$$

In this concrete example all listed pairs are distinct, so $m_L(e_i, e_j) = 1$ for the six pairs above and $m_L(x, y) = 0$ otherwise. Operationally, a vertex of $L_{\mathrm{md}}(G)$ represents a shipment, and an edge of $L_{\mathrm{md}}(G)$ represents a feasible two-leg itinerary where the destination of the first shipment equals the origin of the second (including chaining through the loop e_4 at depot B).

Theorem 6.5 (Well-definedness)**.** *For every finite multidirected graph $G = (V, E, s, t, m)$, the structure $L_{\mathrm{md}}(G) = (V_L, E_L, s_L, t_L, m_L)$ of Definition 6.3 is a finite multidirected graph.*

Proof. Finiteness is immediate because $V_L = E$ and $E_L \subseteq E \times E$ are finite. By construction, $s_L, t_L : E_L \to V_L$ are well-defined maps, and $m_L : V_L \times V_L \to \mathbb{N}_0$ counts (possibly zero) how many edges of E_L go from one line-vertex to another. Hence $L_{\mathrm{md}}(G)$ satisfies Definition **??**. □

Theorem 6.6 (Generalizes the line directed graph)**.** *Let* $D = (V, A)$ *be a (multi)digraph, and view it as a multidirected graph by* $G_D = (V, E, s, t, m)$ *with* $E = A$ *and* $m(u, v) = \#\{\, a \in A : s(a) = u,\ t(a) = v \,\}$*. Then the underlying digraph of* $L_{\mathrm{md}}(G_D)$ *is canonically isomorphic to the classical line directed graph* $L(D)$*: its vertices are the arcs of D, and there is an arc* $(a_1 \longrightarrow a_2)$ *iff* $t(a_1) = s(a_2)$*.*

Proof. In $L_{\mathrm{md}}(G_D)$ we have $V_L = E = A$. By definition, $E_L = \{(a_1, a_2) \in A \times A :\ t(a_1) = s(a_2)\}$, and the source/target maps are $s_L(a_1, a_2) = a_1$ and $t_L(a_1, a_2) = a_2$. Forgetting the multiplicity bookkeeping recovers exactly the arc set of the line digraph $L(D)$. ◻

Definition 6.7 (Orientation functor Φ)**.** Let $G_{\mathrm{bi}} = (V, E, \tau)$ be an *orientable* bidirected graph, i.e., each $e \in E$ is incident with exactly two vertices u, v and $\{\tau(u, e), \tau(v, e)\} = \{-1, +1\}$. Define a multidirected graph

$$\Phi(G_{\mathrm{bi}}) = (V, E, s, t, m)$$

by setting $s(e)$ to be the unique vertex with $\tau(s(e), e) = -1$, and $t(e)$ the unique vertex with $\tau(t(e), e) = +1$, while $m(u, v)$ again counts parallel edges.

Theorem 6.8 (Generalizes the line bidirected graph (orientable case))**.** *Let* G_{bi} *be an orientable bidirected graph and let* $\Phi(G_{\mathrm{bi}})$ *be as in Definition 6.7. Consider the underlying (unsigned) digraph of the line bidirected graph* $L_{\mathrm{bi}}(G_{\mathrm{bi}})$ *obtained by directing each "turn" from* e_{in} *to* e_{out} *and forgetting endpoint signs. Then there is a natural digraph isomorphism*

$$U(L_{\mathrm{bi}}(G_{\mathrm{bi}})) \ \cong\ \textit{underlying digraph of } L_{\mathrm{md}}(\Phi(G_{\mathrm{bi}})).$$

Proof. Vertices on both sides are the edges E of the original structure. (1) If in $L_{\mathrm{bi}}(G_{\mathrm{bi}})$ there is a turn $e_{\mathrm{in}} \longrightarrow e_{\mathrm{out}}$ at v with $e_{\mathrm{in}} \in E^+(v)$ and $e_{\mathrm{out}} \in E^-(v)$, then by the definition of Φ, $t(e_{\mathrm{in}}) = v = s(e_{\mathrm{out}})$. Hence $(e_{\mathrm{in}}, e_{\mathrm{out}}) \in E_L$ of $L_{\mathrm{md}}(\Phi(G_{\mathrm{bi}}))$, i.e., there is a line edge $e_{\mathrm{in}} \longrightarrow e_{\mathrm{out}}$. (2) Conversely, if $(e_1, e_2) \in E_L$ of $L_{\mathrm{md}}(\Phi(G_{\mathrm{bi}}))$, then $t(e_1) = s(e_2) =: v$. In the bidirected signs, this means $\tau(v, e_1) = +1$ and $\tau(v, e_2) = -1$, hence there is a turn $e_1 \longrightarrow e_2$ at v in $L_{\mathrm{bi}}(G_{\mathrm{bi}})$. Thus we have a bijection on arcs preserving incidence, which gives the claimed isomorphism. ◻

7 Conclusion

In this paper, we examined the concepts of line graphs and total graphs in the contexts of *Directed HyperGraphs*, *Directed SuperHyperGraphs*, *Bidirected Graphs*, and *Multidirected Graphs*. We also provided illustrative examples and briefly discussed some of their mathematical properties.

Looking ahead, we expect that the concepts introduced here will be further enriched by incorporating advanced frameworks such as the Fuzzy Set [83, 84], Intuitionistic Fuzzy Set [85, 86], Neutrosophic Set [87–89], HyperFuzzy Set [90–93], Soft Set [94, 95], HyperSoft Set [96, 97], Complex Fuzzy Set [98–100], Rough Set [101–103], Hesitant Fuzzy Set [104, 105], Picture Fuzzy Set [106, 107], Plithogenic Set [108–110], and Quadripartitioned Neutrosophic Set [111–113], among others.

We also anticipate progress in the design of graph algorithms and the validation of these concepts through computational experiments, as well as the exploration of deeper mathematical structures underlying these frameworks.

Funding

No external funding was received for this work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this work.

Acknowledgments

We thank all colleagues, reviewers, and readers whose comments and questions have greatly improved this manuscript. We are also grateful to the authors of the works cited herein for providing the theoretical foundations that underpin our study. Finally, we appreciate the institutional and technical support that enabled this research.

Data Availability

This paper is theoretical and did not generate or analyze any empirical data. We welcome future studies that apply and test these concepts in practical settings.

Research Integrity

The author confirms that this manuscript is original, has not been published elsewhere, and is not under consideration by any other journal.

Use of Computational Tools

All proofs and derivations were performed manually; no computational software (e.g., Mathematica, SageMath, Coq) was used.

Code Availability

No code or software was developed for this study.

Ethical Approval

This research did not involve human participants or animals, and therefore did not require ethical approval.

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.

Supplementary Information

No supplementary materials accompany this paper.

Disclaimer

The ideas presented here are theoretical and have not yet been validated through empirical testing. While we have strived for accuracy and proper citation, inadvertent errors may remain. Readers should verify any referenced material independently. The opinions expressed are those of the authors and do not necessarily reflect the views of their institutions.

References

[1] M Amin Bahmanian and Mateja Šajna. Hypergraphs: connection and separation. *arXiv preprint arXiv:1504.04274*, 2015.

[2] Mohammad A Bahmanian and Mateja Sajna. Connection and separation in hypergraphs. *Theory and Applications of Graphs*, 2(2):5, 2015.

[3] 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.

[4] Mohammad Hamidi, Marzieh Rahmati, and Akbar Rezaei. *Boolean expression based on hypergraphs with algorithm*. Infinite Study, 2021.

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

[6] Junwu Chen and Philippe Schwaller. Molecular hypergraph neural networks. *The Journal of Chemical Physics*, 160(14), 2024.

[7] Yuxin Wang, Quan Gan, Xipeng Qiu, Xuanjing Huang, and David Wipf. From hypergraph energy functions to hypergraph neural networks. In *International Conference on Machine Learning*, pages 35605–35623. PMLR, 2023.

[8] Bo-Lan Wang, Guibao Shen, Dong Li, Jianye Hao, Wulong Liu, Yu Huang, Hongzhong Wu, Yibo Lin, Guangyong Chen, and Pheng-Ann Heng. Lhnn: lattice hypergraph neural network for vlsi congestion prediction. *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022.

[9] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. *ArXiv*, abs/2106.13264, 2021.

[10] Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. *ArXiv*, abs/2105.00956, 2021.

[11] Waheed Ahmad Khan, Waqar Arif, Hossein Rashmanlou, and Saeed Kosari. Interval-valued picture fuzzy hypergraphs with application towards decision making. *Journal of Applied Mathematics and Computing*, 70(2):1103–1125, 2024.

[12] Zengtai Gong and Junhu Wang. Hesitant fuzzy graphs, hesitant fuzzy hypergraphs and fuzzy graph decisions. *Journal of Intelligent & Fuzzy Systems*, 40(1):865–875, 2021.

[13] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pages 3183–3191. PMLR, 2019.

[14] Junyoung Park, Fangying Chen, and Jinkyoo Park. A molecular hypergraph convolutional network with functional group information. 2021.

[15] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. *HyperSoft Set Methods in Engineering*, 3:32–61, 2025.

[16] 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.

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

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

[19] Takaaki Fujita. Exploration of graph classes and concepts for superhypergraphs and n-th power mathematical structures. 2025.

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

[21] Takaaki Fujita. Review of artificial neural hypernetworks and superhypernetworks based on hypergraphs, superhypergraphs, and powersets. 2025.

[22] Takaaki Fujita. Superhypergraph neural network and dynamic superhypergraph neural network. *Authorea Preprints*, 2025.

[23] Takaaki Fujita. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. *arXiv preprint arXiv:2412.01176*, 2024.

[24] Takaaki Fujita. Modeling hierarchical systems in graph signal processing, electric circuits, and bond graphs via hypergraphs and superhypergraphs. *Journal of Engineering Research and Reports*, 27(5):542, 2025.

[25] Julio Cesar Méndez Bravo, Claudia Jeaneth Bolanos Piedrahita, Manuel Alberto Méndez Bravo, and Luis Manuel Pilacuan-Bonete. Integrating smed and industry 4.0 to optimize processes with plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 84(1):27, 2025.

[26] Lisbeth J Reales-Chacón, María E Lucena de Ustáriz, Francisco J Ustáriz-Fajardo, Andrea C Peñafiel Luna, Gabriela J Bonilla-Ayala, Pablo Djabayan-Djibeyan, José L Erazo-Parra, and Mónica A Valdiviezo-Maygua. Study of the efficacy of neural mobilizations to improve sensory and functional responses of lower extremities in older adults with diabetic peripheral neuropathy using plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 74:1–12, 2024.

[27] 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.

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

[29] Mohammad Hamidi and Mohadeseh Taghinezhad. *Application of Superhypergraphs-Based Domination Number in Real World*. Infinite Study, 2023.

[30] Takaaki Fujita. Rethinking strategic perception: Foundations and advancements in hypergame theory and superhypergame theory. *Prospects for Applied Mathematics and Data Analysis*, 4(2):01–14, 2024.

[31] Karel Hrbacek and Thomas Jech. *Introduction to set theory, revised and expanded*. Crc Press, 2017.

[32] Robert L Vaught. *Set theory: an introduction*. Springer Science & Business Media, 2001.

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

[34] Florentin Smarandache. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*. Infinite Study, 2023.

[35] Florentin Smarandache. Extension of hyperalgebra to superhyperalgebra and neutrosophic superhyperalgebra (revisited). In *International Conference on Computers Communications and Control*, pages 427–432. Springer, 2022.

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

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

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

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

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

[41] 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.

[42] Jamie Burke, Ashley Akbari, Rowena Bailey, Kevin Fasusi, Ronan A. Lyons, Jonathan Pearson, James Rafferty, and Daniel Schofield. Representing multimorbid disease progressions using directed hypergraphs. In *medRxiv*, 2023.

[43] Sunwoo Kim, Minyoung Choe, Jaemin Yoo, and Kijung Shin. Reciprocity in directed hypergraphs: measures, findings, and generators. *Data Mining and Knowledge Discovery*, 37:2330–2388, 2022.

[44] Kazusato Oko, Shinsaku Sakaue, and Shin ichi Tanigawa. Nearly tight spectral sparsification of directed hypergraphs by a simple iterative sampling algorithm. *ArXiv*, abs/2204.02537, 2022.

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

[46] K. K. Myithili, Rangasamy Parvathi, and Muhammad Akram. Certain types of intuitionistic fuzzy directed hypergraphs. *International Journal of Machine Learning and Cybernetics*, 7:287–295, 2016.

[47] Yuri V. Muranov, Anna Szczepkowska, and Vladimir Valentinovich Vershinin. Homology of weighted path complexes and directed hypergraphs. *Portugaliae Mathematica*, 2022.

[48] Mauro Faccin. Measuring dynamical systems on directed hypergraphs. *Physical review. E*, 106 3-1:034306, 2022.

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

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

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

[52] Takaaki Fujita. Directed acyclic superhypergraphs (dash): A general framework for hierarchical dependency modeling. *Neutrosophic Knowledge*, 6:72–86, 2025.

[53] Laithun Boro, Madan Mohan Singh, and Jituparna Goswami. Line graph associated to the intersection graph of ideals of rings. *J. Math. Comput. Sci.*, 11(3):2736–2745, 2021.

[54] DVS Sastry and B Syam Prasad Raju. Graph equations for line graphs, total graphs, middle graphs and quasi-total graphs. *Discrete Mathematics*, 48(1):113–119, 1984.

[55] Douglas Bauer and Ralph Tindell. The connectivities of line and total graphs. *Journal of Graph Theory*, 6(2):197–203, 1982.

[56] Quentin Japhet, Dimitri Watel, Dominique Barth, and Marc-Antoine Weisser. Maximal line digraphs. *arXiv preprint arXiv:2406.05141*, 2024.

[57] Lowell W Beineke and Jay S Bagga. Super line graphs and super line digraphs. In *Line graphs and line digraphs*, pages 233–256. Springer, 2021.

[58] Cristina Dalfó and MA Fiol. Cospectral digraphs from locally line digraphs. *Linear Algebra and its Applications*, 500:52–62, 2016.

[59] Daniela Ferrero. Some properties of superline digraphs. In *Fourth International Conference on Information Technology (ITNG'07)*, pages 1073–1076. IEEE, 2007.

[60] Toru Hasunuma. Queue layouts of iterated line directed graphs. *Discrete applied mathematics*, 155(9):1141–1154, 2007.

[61] Martin Knor and L'udovit Niepel. Connectivity of iterated line graphs. *Discrete applied mathematics*, 125(2-3):255–266, 2003.

[62] Toru Hasunuma. Embedding iterated line digraphs in books. *Networks: An International Journal*, 40(2):51–62, 2002.

[63] Martin Knor and L'udovit Niepel. Iterated line graphs are maximally ordered. *Journal of Graph Theory*, 52(2):171–180, 2006.

[64] Toru Hasunuma and Mayu Otani. On the (h, k)-domination numbers of iterated line digraphs. *Discrete Applied Mathematics*, 160(12):1859–1863, 2012.

[65] Toru Hasunuma. Laying out iterated line digraphs using queues. In *International Symposium on Graph Drawing*, pages 202–213. Springer, 2003.

[66] Yehong Shao. Connectivity of iterated line graphs. *Discrete applied mathematics*, 158(18):2081–2087, 2010.

[67] Camino Balbuena, Xavier Marcote, and Daniela Ferrero. Diameter vulnerability of iterated line digraphs in terms of the girth. *Networks: An International Journal*, 45(2):49–54, 2005.

[68] Lowell W Beineke and Jay S Bagga. Iterated line digraphs. In *Line Graphs and Line Digraphs*, pages 173–199. Springer, 2021.

[69] Gary Chartrand and M James Stewart. Total digraphs. *Canadian Mathematical Bulletin*, 9(2):171–176, 1966.

[70] Lowell W Beineke and Jay S Bagga. Total graphs and total digraphs. In *Line Graphs and Line Digraphs*, pages 203–217. Springer, 2021.

[71] Zhenhua Liu and Zhao Zhang. Restricted connectivity of total digraph. *Discrete Mathematics, Algorithms and Applications*, 8(02):1650022, 2016.

[72] MC Kumar and HM Nagesh. Directed pathos total digraph of an arborescence. *Engineering and Applied Science Letters*, 1(1):29–42, 2018.

[73] Jill Maegan B Pamplona and Imelda S Aniversario. On the planarity of a directed pathos total digraphs of some special arborescence graphs. *European Journal of Pure and Applied Mathematics*, 15(3):1331–1343, 2022.

[74] Nanao Kita. Bidirected graphs i: Signed general kotzig-lovász decomposition. *arXiv: Combinatorics*, 2017.

[75] Rui Xu and Cun-Quan Zhang. On flows in bidirected graphs. *Discrete mathematics*, 299(1-3):335–343, 2005.

[76] Jes'us Arturo Jim'enez Gonz'alez and Andrzej Mr'oz. Bidirected graphs, integral quadratic forms and some diophantine equations. 2023.

[77] Laura Gellert and Raman Sanyal. On degree sequences of undirected, directed, and bidirected graphs. *European Journal of Combinatorics*, 64:113–124, 2017.

[78] André Bouchet. Nowhere-zero integral flows on a bidirected graph. *Journal of Combinatorial Theory, Series B*, 34(3):279–292, 1983.

[79] Sebastian Pardo-Guerra, Vivek Kurien George, and Gabriel A Silva. On the graph isomorphism completeness of directed and multidirected graphs. *Mathematics (2227-7390)*, 13(2), 2025.

[80] Takaaki Fujita. Extensions of multidirected graphs: Fuzzy, neutrosophic, plithogenic, rough, soft, hypergraph, and superhypergraph variants. *International Journal of Topology*, 2(3):11, 2025.

[81] Takaaki Fujita. Review of bidirected graph neural networks and multidirected graph neural networks. *Authorea Preprints*, 2025.

[82] Sebastian Pardo-Guerra, Vivek Kurien George, Vikash Morar, Joshua Roldan, and Gabriel Alex Silva. Extending undirected graph techniques to directed graphs via category theory. *Mathematics*, 12(9):1357, 2024.

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

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

[85] Krassimir T Atanassov and G Gargov. *Intuitionistic fuzzy logics*. Springer, 2017.

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

[87] Madeleine Al Tahan, Saba Al-Kaseasbeh, and Bijan Davvaz. Neutrosophic quadruple hv-modules and their fundamental module. *Neutrosophic Sets and Systems*, 72:304–325, 2024.

[88] Said Broumi, Assia Bakali, Mohamed Talea, and Florentin Smarandache. An isolated bipolar single-valued neutrosophic graphs. In *Information Systems Design and Intelligent Applications: Proceedings of Fourth International Conference INDIA 2017*, pages 816–822. Springer, 2018.

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

[90] Jayanta Ghosh and Tapas Kumar Samanta. Hyperfuzzy sets and hyperfuzzy group. *Int. J. Adv. Sci. Technol*, 41:27–37, 2012.

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

[92] Young Bae Jun, Seok-Zun Song, and Seon Jeong Kim. Distances between hyper structures and length fuzzy ideals of bck/bci-algebras based on hyper structures. *Journal of Intelligent & Fuzzy Systems*, 35(2):2257–2268, 2018.

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

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

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

[96] Sagvan Y Musa and Baravan A Asaad. Topological structures via bipolar hypersoft sets. *Journal of Mathematics*, 2022(1):2896053, 2022.

[97] Florentin Smarandache. Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic sets and systems*, 22(1):168–170, 2018.

[98] Tahir Mahmood, Ubaid ur Rehman, Abdul Jaleel, Jabbar Ahmmad, and Ronnason Chinram. Bipolar complex fuzzy soft sets and their applications in decision-making. *Mathematics*, 2022.

[99] Tahir Mahmood and Ubaid ur Rehman. A novel approach towards bipolar complex fuzzy sets and their applications in generalized similarity measures. *International Journal of Intelligent Systems*, 37:535 – 567, 2021.

[100] Daniel Ramot, Menahem Friedman, Gideon Langholz, Ron Milo, and Abraham Kandel. On complex fuzzy sets. *10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297)*, 3:1160–1163 vol.2, 2001.

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

[102] Zdzis law Pawlak and Andrzej Skowron. Rudiments of rough sets. *Information sciences*, 177(1):3–27, 2007.

[103] Said Broumi, Florentin Smarandache, and Mamoni Dhar. Rough neutrosophic sets. *Infinite Study*, 32:493–502, 2014.

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

[105] Zeshui Xu. *Hesitant fuzzy sets theory*, volume 314. Springer, 2014.

[106] Sankar Das, Ganesh Ghorai, and Madhumangal Pal. Picture fuzzy tolerance graphs with application. *Complex & Intelligent Systems*, 8(1):541–554, 2022.

[107] 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.

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

[109] WB Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. *Plithogenic Graphs*. Infinite Study, 2020.

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

[111] Satham Hussain, Jahir Hussain, Isnaini Rosyida, and Said Broumi. Quadripartitioned neutrosophic soft graphs. In *Handbook of Research on Advances and Applications of Fuzzy Sets and Logic*, pages 771–795. IGI Global, 2022.

[112] S Satham Hussain, N Durga, Muhammad Aslam, G Muhiuddin, and Ganesh Ghorai. New concepts on quadripartitioned neutrosophic competition graph with application. *International Journal of Applied and Computational Mathematics*, 10(2):57, 2024.

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