Article

Not peer-reviewed version

HyperMatrix, SuperHyperMatrix, MultiMatrix, Iterative MultiMatrix, MetaMatrix, and Iterated MetaMatrix

Takaaki Fujita *

Posted Date: 3 October 2025

doi: 10.20944/preprints202510.0232.v1

Keywords: hyperstructure; superhyperstructure; multi-structure; iterative multi-structure; matrix

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

Article

HyperMatrix, SuperHyperMatrix, MultiMatrix, Iterative MultiMatrix, MetaMatrix, and Iterated MetaMatrix

Takaaki Fujita

Independent Researcher, Tokyo, Japan; takaaki.fujita060@gmail.com

Abstract

We begin with the classical viewpoint in which a *Structure* consists of a nonempty carrier together with single–valued basic operations. A *Hyperstructure* arises by promoting operations to act on (and return) subsets of a base set, i.e., on its powerset. Iterating the powerset operator $\mathcal{P}$ n times yields an *n-Superhyperstructure*: informally, the n-th powerset $\mathcal{P}^n(S)$ is obtained by n successive applications of $\mathcal{P}$ (cf. [1]). We review the fundamental definitions and give compact, instructive examples. A *Multi-Structure* replaces classical operations with maps from tuples to finite multisets, thereby allowing multiple outputs per input in a controlled, simultaneous manner. A *MetaStructure* treats whole structures as elements and equips them with uniform, isomorphism–invariant operations that functorially construct new structures from existing ones. In this paper we define *HyperMatrix*, *SuperHyperMatrix*, *MultiMatrix*, *Iterative MultiMatrix*, *MetaMatrix*, and *Iterated MetaMatrix*—all as extensions of the classical notion of a matrix—and we offer a concise examination of their properties.

Keywords: hyperstructure; superhyperstructure; multi-structure; iterative multi-structure; matrix

---

1. Preliminaries

This section gathers the basic notions used throughout the paper. Unless stated otherwise, all sets are taken to be finite.

1.1. Classical Structures, Hyperstructures, and n-Superhyperstructures

A *Classical Structure* is an ordinary algebraic/relational system on a single carrier. A *Hyperstructure* is obtained by promoting outcomes of operations to *sets* (via the powerset). Iterating the powerset n times gives rise to an *n-Superhyperstructure*; see, e.g., [1–10]. Intuitively, the n-fold powerset records n layers of "grouping" or aggregation.

Definition 1 (Base set)**.** *A* base set *is a nonempty collection S of atomic elements from which we form derived objects such as $\mathcal{P}(S)$ and the iterated powersets $\mathcal{P}^n(S)$. Formally,*

$$S \;=\; \{\, x \mid x \text{ belongs to the fixed universe under study} \,\}.$$

Definition 2 (Powerset)**.** *[11] For any set S, the* powerset *$\mathcal{P}(S)$ is the set of all subsets of S:*

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

Definition 3 (Iterated (nonempty) powersets)**.** *(cf. [1,12,13]) Define $\mathcal{P}^0(H) := H$ and inductively*

$$\mathcal{P}^{k+1}(H) \;:=\; \mathcal{P}\big(\mathcal{P}^k(H)\big) \qquad (k \geq 0).$$

Thus $\mathcal{P}^1(H) = \mathcal{P}(H)$, $\mathcal{P}^2(H) = \mathcal{P}(\mathcal{P}(H))$, and so on. If one wishes to exclude the empty set at each stage, write $\mathcal{P}^*(X) := \mathcal{P}(X) \setminus \{\emptyset\}$ and set $\mathcal{P}^{*0}(H) := H$, $\mathcal{P}^{*(k+1)}(H) := \mathcal{P}^*\big(\mathcal{P}^{*k}(H)\big)$.

 doi:10.20944/preprints202510.0232.v1

Example 1 (Iterated (nonempty) powersets — "Committees and Meeting Days"). *Let the employee set be*

$$H = \{\text{Saki, Ayame, Taro}\} \quad (|H| = 3).$$

The first nonempty powerset $\mathcal{P}^{*1}(H) = \mathcal{P}(H) \setminus \{\emptyset\}$ *consists of all nonempty* committees. *Its cardinality is*

$$|\mathcal{P}^{*1}(H)| = 2^{|H|} - 1 = 2^3 - 1 = 7.$$

The second nonempty powerset $\mathcal{P}^{*2}(H) = \mathcal{P}^*(\mathcal{P}^{*1}(H))$ *collects all nonempty* meeting-day plans, *i.e., nonempty families of committees. Its cardinality is*

$$|\mathcal{P}^{*2}(H)| = 2^{|\mathcal{P}^{*1}(H)|} - 1 = 2^7 - 1 = 127.$$

A concrete meeting-day plan is

$$\mathcal{D} = \{\{\text{Saki, Ayame}\}, \{\text{Taro}\}\} \in \mathcal{P}^{*2}(H),$$

interpreted as: on that day, the $\{\text{Saki, Ayame}\}$ *committee and the* $\{\text{Taro}\}$ *committee both convene.*

Definition 4 (Classical Structure). *[14] A* Classical Structure *is a pair*

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

where $H \neq \emptyset$ *is the carrier and each* $\#^{(m)} : H^m \to H$ *(for* $m \in \mathcal{I} \subseteq \mathbb{Z}_{>0}$*) is a single–valued basic operation subject to the axioms appropriate to the intended theory. Typical instances include:*

- Sets and logics: *a set with designated relations; propositional algebras* $(L, \wedge, \vee, \neg)$.
- Measure/probability: $(\Omega, \mathcal{F}, P)$ *with* $P : \mathcal{F} \to [0, 1]$.
- Algebra: *groups, rings, and vector spaces with their standard operations [15–18].*
- Geometry/graphs/automata/games: *metric spaces, (di)graphs [19–21], finite automata, and strategic–form games.*

Definition 5 (Hyperoperation). *(cf. [22–25]) Given a set S, a* hyperoperation *is a set–valued binary map*

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

Hence combining two elements can yield a set *of possible outcomes.*

Definition 6 (Hyperstructure). *(cf. [1,13,26–28]) A* Hyperstructure *is a pair* $\mathcal{H} = (\mathcal{P}(S), \circ)$ *in which the basic operation(s) act on (and return) subsets of a base set S. In contrast with classical structures, the output of an operation need not be a single element but may be a whole subset of S.*

Example 2 (Hyperstructure — "Adding Measurements with Bounded Error"). *Let the base set be real lengths in centimeters,* $S = \mathbb{R}$. *Fix a worst-case device error bound* $\Delta = 0.20$ *cm (e.g., two instruments each with* ± 0.10 *cm). Define the hyperoperation*

$$x \circ y := [\, x + y - \Delta,\ x + y + \Delta \,] \subseteq \mathbb{R} \quad (x, y \in S).$$

This yields the hyperstructure $\mathcal{H} = (\mathcal{P}(S), \circ)$*, where combining two measured values returns the* set *of all physically plausible sums given the error. Numerical instance:*

$$12.30 \circ 7.90 = [\, 12.30 + 7.90 - 0.20,\ 12.30 + 7.90 + 0.20 \,] = [\, 20.00,\ 20.40 \,].$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

If a third measurement 5.00 *(same error model) is combined, the set-wise extension gives*

$$(12.30 \circ 7.90) \circ 5.00 = \bigcup_{u \in [20.00, 20.40]} [\, u + 5.00 - \Delta,\ u + 5.00 + \Delta\,] = [\,25.00,\ 25.60\,],$$

again a subset of S.

Definition 7 (SuperHyperOperations)**.** *(cf. [1]) Let* $H \neq \emptyset$ *and let* $\mathcal{P}^n(H)$ *be as in Definition 3. An* (m, n)-SuperHyperOperation *is an* m*-ary map*

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

where $\mathcal{P}_*^{\,n}(H)$ *denotes either the full* n*-th powerset or its nonempty variant. Allowing* $n \geq 1$ *permits set–valued outputs (and, for* $n \geq 2$*, nested families of sets).*

Definition 8 (n-Superhyperstructure)**.** *(cf. [1,5,13]) For a base set* S *and* $n \geq 1$*, an* n-Superhyperstructure *is any system*

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

whose operations are defined on the n*-fold powerset. The case* $n = 1$ *recovers hyperstructures, while larger* n *encode multi-level aggregation.*

Example 3 (n-Superhyperstructure ($n = 2$) — "Cross-Functional Project Plans")**.** *Let* $S = \{\text{Design, Build, Test}\}$ *be atomic tasks. Then* $\mathcal{P}(S)$ *are* teams *(task bundles), and* $\mathcal{P}^2(S) = \mathcal{P}(\mathcal{P}(S))$ *are* project plans *(families of teams). Define an operation* $\diamond : \mathcal{P}^2(S) \times \mathcal{P}^2(S) \to \mathcal{P}^2(S)$ *by*

$$\mathcal{A} \diamond \mathcal{B} \ := \ \{\, A \cup B \ :\ A \in \mathcal{A},\ B \in \mathcal{B}\,\}.$$

Concrete data:

$$\mathcal{A} = \{\{\text{Design}, \text{Build}\}\}, \qquad \mathcal{B} = \{\{\text{Build}, \text{Test}\},\ \{\text{Design}\}\}.$$

Then

$$\mathcal{A} \diamond \mathcal{B} = \{\{\text{Design}, \text{Build}, \text{Test}\},\ \{\text{Design}, \text{Build}\}\} \ \in \ \mathcal{P}^2(S),$$

which aggregates two plans into a new plan by pairwise union of teams—typical of multi-team synchronization. Thus $\mathcal{SH}_2 = (\mathcal{P}^2(S), \diamond)$ *is a 2-superhyperstructure.*

Definition 9 ((m, n)-SuperHyperStructure)**.** *(cf. [12,29]) Let* $S \neq \emptyset$ *and* $0 \leq m \leq n$*. An* (m, n)-SuperHyperStructure *(of arity* s*) consists of an operation*

$$\odot^{(m,n)} :\ \big(\mathcal{P}^{\,m}(S)\big)^s \longrightarrow \mathcal{P}^{\,n}(S).$$

Specializations include ordinary s*-ary operations when* $m = n = 0$*, hyperoperations when* $(m, n) = (0, 1)$*, and superhyperoperations when* $s = 1$*. Thus the* (m, n)*-formalism uniformly bridges classical, hyper, and higher-level set-valued behaviors.*

Example 4 ((m, n)-SuperHyperStructure ($m = 1,\ n = 2,\ s = 2$) — "Shipping Bundle Alternatives")**.** *Let the item universe be* $S = \{\text{Oats, Milk, Bread, Eggs}\}$*. Elements of* $\mathcal{P}^1(S) = \mathcal{P}(S)$ *are* carts *(chosen items), while elements of* $\mathcal{P}^2(S)$ *are* bundle families *(alternative packagings made of item-bundles). Define*

$$\odot^{(1,2)} :\ \mathcal{P}(S) \times \mathcal{P}(S) \longrightarrow \mathcal{P}^2(S), \qquad (A, B) \ \mapsto \ \{\ A \cup B,\ \ \{A, B\}\ \}.$$

Here $A \cup B$ *is the "single-box kit" alternative, and* $\{A, B\}$ *is the "two-box split" alternative. Take*

$$A = \{\text{Oats}, \text{Milk}\}, \qquad B = \{\text{Bread}\}.$$

 doi:10.20944/preprints202510.0232.v1

Then

$$\odot^{(1,2)}(A,B) = \Big\{ \{\text{Oats}, \text{Milk}, \text{Bread}\},\ \{\{\text{Oats}, \text{Milk}\}, \{\text{Bread}\}\} \Big\} \in \mathcal{P}^2(S).$$

Thus $(\mathcal{P}^1(S), \mathcal{P}^2(S), \odot^{(1,2)})$ *realizes an* (m,n)*-SuperHyperStructure that maps two carts to a family of feasible shipping-packaging plans.*

1.2. Multi-Structure

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

Definition 10 (Finite Multiset). *(cf.[30–33]) Let H be a nonempty set. A* finite multiset *on H is a function*

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

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

Definition 11 (MultiOperation). *Let H be a nonempty set and fix an integer* $m \geq 1$*. A* multi-operation *of arity m on H is a map*

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

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

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

Example 5 (MultiOperation — "Frequently bought together" in retail). *Let H be the set of store SKUs*

$$H = \{Bread,\ Butter,\ Milk,\ Eggs,\ Jam\}.$$

Define a binary multi-operation $\#^{(2)} : H^2 \to \mathcal{M}(H)$ *that returns a* finite multiset of recommended companion items*; multiplicities encode strength or quantity:*

$$\#^{(2)}(x,y) = \begin{cases} \{\{Butter^2,\ Milk^1\}\}, & \text{if } \{x,y\} = \{Bread, Eggs\}, \\ \{\{Jam^2\}\}, & \text{if } \{x,y\} = \{Bread, Butter\}, \\ \{\{Eggs^1\}\}, & \text{if } \{x,y\} = \{Milk, Bread\}, \\ \varnothing, & \text{otherwise.} \end{cases}$$

For instance, from the basket $(Bread, Butter)$ *the system proposes* $\#^{(2)}(Bread, Butter) = \{\{Jam^2\}\}$*, i.e. "suggest two jars of jam." This is a concrete real-world* multi-operation *because the output is a multiset of items in H.*

Definition 12 (MultiStructure). *[34,35] A* MultiStructure *is a pair*

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

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

Example 6 (MultiStructure — unified retail recommendation rules of mixed arity). *Let the carrier be*

$$H = \{Bread,\ Butter,\ Milk,\ Eggs,\ Jam,\ LactoseFreeMilk\}.$$

Define two multi-operations (different arities) that act simultaneously on H:

- ***Unary substitute rule*** $\#^{(1)} : H \to \mathcal{M}(H)$*:*

$$\#^{(1)}(h) = \begin{cases} \{\{LactoseFreeMilk^1\}\}, & \text{if } h = Milk, \\ \{\{Butter^1\}\}, & \text{if } h = Bread, \\ \varnothing, & \text{otherwise.} \end{cases}$$

- ***Binary bundle rule*** $\#^{(2)} : H^2 \to \mathcal{M}(H)$*:*

$$\#^{(2)}(x,y) = \begin{cases} \{\{Jam^2\}\}, & \text{if } \{x,y\} = \{Bread, Butter\}, \\ \{\{Eggs^1,\ Butter^1\}\}, & \text{if } \{x,y\} = \{Bread, Milk\}, \\ \varnothing, & \text{otherwise.} \end{cases}$$

Then

$$\mathcal{MS} = \big(H,\ \{\#^{(1)}, \#^{(2)}\}\big)$$

is a MultiStructure*: it fixes a nonempty carrier H and equips it with a family of multi-operations of various arities. A typical evaluation gives*

$$\#^{(1)}(Milk) = \{\{LactoseFreeMilk^1\}\}, \qquad \#^{(2)}(Bread, Butter) = \{\{Jam^2\}\},$$

showing how single-item substitutions and two-item bundle suggestions coexist in one practical system.

1.3. Iterative Multi-Structure

An Iterative Multi-Structure extends multiset operations across levels, combining multisets of multisets iteratively through k hierarchical stages in layered aggregation [34,35].

Definition 13 (Iterative Multi-Structure of Order k)**.** *[34,35] Let H be a nonempty set and fix an integer* $k \geq 1$*. Define iteratively the* multiset powersets

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

where $\mathcal{M}(X)$ *denotes the collection of finite multisets on X (Definition 10). Let* $\mathcal{I} \subseteq \mathbb{Z}_{>0}$ *index a family of arities. An* Iterative Multi-Structure of order k *is a tuple*

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

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

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

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

Example 7 (Iterative Multi-Structure — meal planning: ingredients → dishes → menus → weekly plan)**.** *Fix depth* $k = 3$*. Let the carrier of* ingredients *be*

$$H = \{\text{Pasta},\ \text{Tomato},\ \text{Basil},\ \text{Cheese}\}.$$

Recall $\mathcal{M}^0(H) = H$*,* $\mathcal{M}^1(H) = \mathcal{M}(H)$ *(finite multisets of ingredients),* $\mathcal{M}^2(H) = \mathcal{M}\big(\mathcal{M}^1(H)\big)$ *(finite multisets of dishes), and* $\mathcal{M}^3(H) = \mathcal{M}\big(\mathcal{M}^2(H)\big)$ *(finite multisets of menus).*

Define levelwise multi-operations of arity 2:

$$\begin{aligned}
\#^{(2,0)} &: H^2 \longrightarrow \mathcal{M}^1(H) && \textit{(assemble a } \text{dish} \textit{ from two ingredients)},\\
\#^{(2,1)} &: \big(\mathcal{M}^1(H)\big)^2 \longrightarrow \mathcal{M}^2(H) && \textit{(assemble a } \text{menu} \textit{ from two dishes)},\\
\#^{(2,2)} &: \big(\mathcal{M}^2(H)\big)^2 \longrightarrow \mathcal{M}^3(H) && \textit{(assemble a } \text{weekly plan} \textit{ from two menus)}.
\end{aligned}$$

Concretely, for $x, y \in H$ put

$$\#^{(2,0)}(x,y) = \begin{cases} \{\{\mathrm{Pasta}^1,\ \mathrm{Tomato}^2\}\}, & \textit{if } \{x,y\} = \{\mathrm{Pasta}, \mathrm{Tomato}\},\\ \{\{\mathrm{Tomato}^1,\ \mathrm{Basil}^1,\ \mathrm{Cheese}^1\}\}, & \textit{if } \{x,y\} = \{\mathrm{Tomato}, \mathrm{Cheese}\},\\ \varnothing, & \textit{otherwise.} \end{cases}$$

For two dishes $D_1, D_2 \in \mathcal{M}^1(H)$, let

$$\#^{(2,1)}(D_1, D_2) \ :=\ \{\{\, D_1^1,\ D_2^1 \,\}\} \ \in\ \mathcal{M}^2(H),$$

i.e., a menu consisting of exactly those two dishes (multiplicity counts repeats). For two menus $M_1, M_2 \in \mathcal{M}^2(H)$, let

$$\#^{(2,2)}(M_1, M_2) \ :=\ \{\{\, M_1^1,\ M_2^1 \,\}\} \ \in\ \mathcal{M}^3(H),$$

i.e., a weekly plan made of two menus.

Concrete run.

$$D_{\mathrm{PT}} := \#^{(2,0)}(\mathrm{Pasta}, \mathrm{Tomato}) = \{\{\mathrm{Pasta}^1,\ \mathrm{Tomato}^2\}\},$$

$$D_{\mathrm{TC}} := \#^{(2,0)}(\mathrm{Tomato}, \mathrm{Cheese}) = \{\{\mathrm{Tomato}^1,\ \mathrm{Basil}^1,\ \mathrm{Cheese}^1\}\}.$$

Build a menu from these dishes:

$$M := \#^{(2,1)}\big(D_{\mathrm{PT}}, D_{\mathrm{TC}}\big) = \{\{\, D_{\mathrm{PT}}^1,\ D_{\mathrm{TC}}^1 \,\}\} \ \in\ \mathcal{M}^2(H).$$

Duplicate the menu to form a simple weekly plan:

$$W := \#^{(2,2)}(M, M) = \{\{\, M^2 \,\}\} \ \in\ \mathcal{M}^3(H).$$

Thus $\big(H, \{\#^{(2,0)}, \#^{(2,1)}, \#^{(2,2)}\}\big)$ *is an* Iterative Multi-Structure of order $k = 3$*: level 0 combines ingredients into dishes; level 1 combines dishes into menus; level 2 combines menus into a weekly plan, all via multiset aggregation.*

1.4. MetaStructure (Structure of Structure)

A MetaStructure organizes structures as elements, providing uniform, isomorphism-invariant operations that construct new structures from existing ones via functorial recipes[36,37].

Notation 1. *Fix a single–sorted, finitary signature* $\Sigma = (\mathsf{Func}, \mathsf{Rel}, \mathsf{ar})$*. A Σ-structure is*

$$\mathbf{C} = \big(H,\ (f^{\mathbf{C}})_{f \in \mathsf{Func}},\ (R^{\mathbf{C}})_{R \in \mathsf{Rel}}\big),$$

with carrier $H \neq \varnothing$, operations $f^{\mathbf{C}} : H^m \to H$ and relations $R^{\mathbf{C}} \subseteq H^r$ of the prescribed arities. Let Str_Σ *be the class of all Σ-structures.*

Definition 14 (MetaStructure over a fixed signature)**.** *[36,37] A* MetaStructure *over Σ is a pair*

$$\mathbb{M} = \big(U,\ (\Phi_\ell)_{\ell \in \Lambda}\big),$$

where $U \subseteq \mathrm{Str}_\Sigma$, $U \neq \emptyset$, *and for each label* ℓ *of meta-arity* $k_\ell \in \mathbb{N}$ *the map* $\Phi_\ell : U^{k_\ell} \to U$ *is specified* uniformly *as follows: there exist constructors*

$$\Gamma_\ell \text{ (for carriers)}, \qquad \Lambda_\ell^f \text{ (for each } f \in \mathsf{Func}), \qquad \Xi_\ell^R \text{ (for each } R \in \mathsf{Rel}),$$

such that for $(\mathbf{C}_1, \dots, \mathbf{C}_{k_\ell}) \in U^{k_\ell}$, *the structure* $\Phi_\ell(\mathbf{C}_1, \dots, \mathbf{C}_{k_\ell})$ *has*

$$\text{carrier } \Gamma_\ell(\mathbf{C}_1, \dots, \mathbf{C}_{k_\ell}), \quad f^{\Phi_\ell(\cdot)} = \Lambda_\ell^f\big(f^{\mathbf{C}_1}, \dots, f^{\mathbf{C}_{k_\ell}}\big), \quad R^{\Phi_\ell(\cdot)} = \Xi_\ell^R\big(R^{\mathbf{C}_1}, \dots, R^{\mathbf{C}_{k_\ell}}\big).$$

Each Φ_ℓ *is* isomorphism-invariant*: isomorphisms of inputs induce an isomorphism of outputs (naturality).*

Remark 1 (Canonical meta-operations)**.** *All are isomorphism-invariant and uniform in* Σ.

- ***Product*** Π *(arity* 2*): carrier* $H_1 \times H_2$*; operations act componentwise; relations are taken as products.*
- ***Disjoint union*** $\uplus$ *(purely relational* Σ*): carrier* $\{1\} \times H_1 \cup \{2\} \times H_2$*; relations are the tagged unions.*
- ***Reduct / Expansion*** *(arity* 1*): forget or add symbols uniformly with prescribed interpretations.*

Example 8 (MetaStructure — composing city transit networks)**.** *Fix the single–sorted relational signature*

$$\Sigma = \big(\mathsf{Func} = \emptyset,\ \mathsf{Rel} = \{\mathrm{Edge}, \mathrm{Air}\},\ \mathrm{ar}(\mathrm{Edge}) = 2,\ \mathrm{ar}(\mathrm{Air}) = 1\big).$$

A Σ*-structure* $\mathbf{C} = (H, \mathrm{Edge}^{\mathbf{C}}, \mathrm{Air}^{\mathbf{C}})$ *represents an urban transit network:* H *is the set of stops/stations,* $\mathrm{Edge} \subseteq H^2$ *is the directed reachability relation (there is a scheduled connection from the first stop to the second), and* $\mathrm{Air} \subseteq H$ *marks airport terminals (intercity gateways).*

Meta–operation (intercity linking). *Define a binary meta–operation* $\Phi_{\mathrm{link}} : U^2 \to U$ *on the class* $U \subseteq \mathrm{Str}_\Sigma$ *of all such city networks by the uniform constructors:*

$$\begin{aligned}
\Gamma_{\mathrm{link}}(\mathbf{C}_1, \mathbf{C}_2) &:= \{1\} \times H_1 \cup \{2\} \times H_2 \quad \text{(tagged disjoint union of stops)},\\
\Xi_{\mathrm{link}}^{\mathrm{Air}}\big(\mathrm{Air}^{\mathbf{C}_1}, \mathrm{Air}^{\mathbf{C}_2}\big) &:= \{1\} \times \mathrm{Air}^{\mathbf{C}_1} \cup \{2\} \times \mathrm{Air}^{\mathbf{C}_2},\\
\Xi_{\mathrm{link}}^{\mathrm{Edge}}\big(\mathrm{Edge}^{\mathbf{C}_1}, \mathrm{Edge}^{\mathbf{C}_2}\big) &:= \underbrace{\{((1,u),(1,v)) : (u,v) \in \mathrm{Edge}^{\mathbf{C}_1}\}}_{\text{intra–city 1}} \cup \underbrace{\{((2,u),(2)v)) : (u,v) \in \mathrm{Edge}^{\mathbf{C}_2}\}}_{\text{intra–city 2}}\\
&\quad \cup \underbrace{\big\{((1,u),(2,v)), ((2,v),(1,u)) :\ u \in \mathrm{Air}^{\mathbf{C}_1},\ v \in \mathrm{Air}^{\mathbf{C}_2}\big\}}_{\text{new bidirectional intercity links}}.
\end{aligned}$$

Thus $\Phi_{\mathrm{link}}(\mathbf{C}_1, \mathbf{C}_2)$ *is the combined region–wide network that keeps all original city connections and adds intercity edges between every pair of airports. This construction is isomorphism–invariant: relabeling the inputs induces a relabeling of the output.*

Concrete instance. *Let* $\mathbf{C}_1 = (H_1, \mathrm{Edge}^{\mathbf{C}_1}, \mathrm{Air}^{\mathbf{C}_1})$ *with* $H_1 = \{a, a'\}$, $\mathrm{Edge}^{\mathbf{C}_1} = \{(a, a')\}$, $\mathrm{Air}^{\mathbf{C}_1} = \{a\}$, *and* $\mathbf{C}_2 = (H_2, \mathrm{Edge}^{\mathbf{C}_2}, \mathrm{Air}^{\mathbf{C}_2})$ *with* $H_2 = \{b, b'\}$, $\mathrm{Edge}^{\mathbf{C}_2} = \{(b, b')\}$, $\mathrm{Air}^{\mathbf{C}_2} = \{b'\}$. *Then* $\Phi_{\mathrm{link}}(\mathbf{C}_1, \mathbf{C}_2) =: (\widehat{H}, \widehat{\mathrm{Edge}}, \widehat{\mathrm{Air}})$ *has*

$$\widehat{H} = \{(1,a), (1,a'), (2,b), (2,b')\}, \qquad \widehat{\mathrm{Air}} = \{(1,a), (2,b')\},$$

$$\widehat{\mathrm{Edge}} = \{((1,a),(1,a')),\ ((2,b),(2,b'))\} \cup \{((1,a),(2,b')),\ ((2,b'),(1,a))\}.$$

Interpretation: the regional network contains the original city routes $a \to a'$ *and* $b \to b'$*, and adds two intercity legs between the airports* a *and* b' *(both directions). This is a real–world MetaStructure: an operation on* structures *(city transit systems) that uniformly yields a new* structure *(a connected multimodal network) by functorial carrier/relations constructors.*

An Iterated MetaStructure recursively applies MetaStructure construction, forming successive layers where structures of structures create deeper hierarchical meta-levels [36,37].

Definition 15 (Iterated MetaStructure of depth t)**.** *[36,37] An* Iterated MetaStructure of depth t *over* Σ *is any MetaStructure* $\mathfrak{M}^{(t)}$ *of height* t*. When* $s < t$*, we* lift *a height-*s *MetaStructure* $\mathfrak{M}^{(s)} = (U^{(s)}, \{\odot_i\}, \{\mathcal{S}_j\})$ *to height* t *by*

$$\iota_{s\to t}: \; U^{(s)} \xrightarrow{\mathsf{U}_\Sigma^{t-s}} U^{(t)} := \mathsf{U}_\Sigma^{t-s}(U^{(s)}),$$

and, for each $\odot_i : (\mathsf{E}_\Sigma^{m_i})^{k_i} \to \mathcal{P}^{n_i}(\mathsf{E}_\Sigma^{n_i})$*, defining its lift*

$$\odot_i^{\uparrow}: \; \left(\mathsf{E}_\Sigma^{m_i+t-s}\right)^{k_i} \longrightarrow \mathcal{P}^{n_i}(\mathsf{E}_\Sigma^{n_i+t-s}), \quad \odot_i^{\uparrow}\left(\mathsf{U}_\Sigma^{t-s}(x_1),\ldots,\mathsf{U}_\Sigma^{t-s}(x_{k_i})\right) := \mathsf{U}_\Sigma^{t-s}\big(\odot_i(x_1,\ldots,x_{k_i})\big),$$

and similarly for relations $\mathcal{S}_j^{\uparrow} := \left(\mathsf{U}_\Sigma^{t-s}\right)^{\times \ell_j}(\mathcal{S}_j)$.

Example 9 (Iterated MetaStructure — multi-tier transportation federation)**.** *Fix the single–sorted relational signature*

$$\Sigma = \big(\mathsf{Func} = \varnothing,\; \mathsf{Rel} = \{\mathrm{Edge}, \mathrm{Term}\},\; \mathrm{ar}(\mathrm{Edge}) = 2,\; \mathrm{ar}(\mathrm{Term}) = 1\big).$$

A Σ*-structure* $\mathbf{C} = (H, \mathrm{Edge}^{\mathbf{C}}, \mathrm{Term}^{\mathbf{C}})$ *models a transit fragment:* H *is a set of stops,* $\mathrm{Edge} \subseteq H^2$ *is the directed connectivity relation (scheduled links), and* $\mathrm{Term} \subseteq H$ *are designated terminals (hubs).*

Base level (0). *Let* $U^{(0)} \subseteq \mathrm{Str}_\Sigma$ *be the class of* local lines *(e.g. bus or metro lines). For a finite family* $(\mathbf{C}_i)_{i\in I} \subset U^{(0)}$*, define the k-ary meta-operation* $\Phi_{\mathrm{link}} : (U^{(0)})^k \to U^{(0)}$ *by the uniform constructors*

$$\begin{aligned}
\Gamma_{\mathrm{link}}\big((\mathbf{C}_i)_{i\in I}\big) &:= \bigsqcup_{i\in I} \{i\}\times H_i \quad \textit{(tagged disjoint union of carriers)},\\
\Xi_{\mathrm{link}}^{\mathrm{Term}}\big((\mathrm{Term}^{\mathbf{C}_i})_{i\in I}\big) &:= \bigcup_{i\in I} \{i\}\times \mathrm{Term}^{\mathbf{C}_i},\\
\Xi_{\mathrm{link}}^{\mathrm{Edge}}\big((\mathrm{Edge}^{\mathbf{C}_i})_{i\in I}\big) &:= \Big(\bigcup_{i\in I} \{i\}\times \mathrm{Edge}^{\mathbf{C}_i}\Big) \cup \Big\{ \big((i,u),(j,v)\big), \big((j,v),(i,u)\big) \\
&\qquad : \; i \neq j,\; u \in \mathrm{Term}^{\mathbf{C}_i},\; v \in \mathrm{Term}^{\mathbf{C}_j} \Big\}.
\end{aligned}$$

Thus Φ_{link} *fuses several lines into a* city network *by tagged union plus bidirectional inter-line links between terminals.*

Iterated lift. *Let* U_Σ *be the canonical "tagging" functor that sends* $\mathbf{C} = (H, \mathrm{Edge}, \mathrm{Term})$ *to*

$$\mathsf{U}_\Sigma(\mathbf{C}) = \big(\{*\}\times H,\; \{*\}\times \mathrm{Edge},\; \{*\}\times \mathrm{Term}\big),$$

and iterate it: U_Σ^r *applies* r *nested tags. Given* $s < t$*, the lifted meta-operation* $\Phi_{\mathrm{link}}^{\uparrow}$ *on height* t *is defined by Definition 15:*

$$\Phi_{\mathrm{link}}^{\uparrow}\big(\mathsf{U}_\Sigma^{t-s}(\mathbf{D}_1),\ldots,\mathsf{U}_\Sigma^{t-s}(\mathbf{D}_k)\big) \;:=\; \mathsf{U}_\Sigma^{t-s}\Big(\, \Phi_{\mathrm{link}}(\mathbf{D}_1,\ldots,\mathbf{D}_k) \,\Big).$$

Intuitively, the same "fuse-and-add-terminal-links" recipe is reused at every tier, while the tags record the tier of origin (city $\to$ *country* $\to$ *region, etc.).*

Concrete 3-tier instance ($t = 3$***).*** *Take three local lines*

$$\mathbf{L}_{A1} = (\{a_1,a_2\},\; \{(a_1,a_2)\},\; \{a_1\}), \quad \mathbf{L}_{A2} = (\{a_3,a_4\},\; \{(a_3,a_4)\},\; \{a_3\}),$$

$$\mathbf{L}_{B1} = (\{b_1,b_2\},\; \{(b_1,b_2)\},\; \{b_2\}).$$

Tier 1 (cities)*. Form two cities by*

$$\mathbf{City}_A := \Phi_{\mathrm{link}}(\mathbf{L}_{A1}, \mathbf{L}_{A2}), \qquad \mathbf{City}_B := \Phi_{\mathrm{link}}(\mathbf{L}_{B1}).$$

Here $\mathbf{City}_A$ *has carrier* $\{1\}\times\{a_1,a_2\} \cup \{2\}\times\{a_3,a_4\}$*, keeps the intra-line edges* $(1,a_1)\to(1,a_2)$*,* $(2,a_3)\to(2,a_4)$*, and adds cross-links* $(1,a_1)\leftrightarrow(2,a_3)$ *between terminals.*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

Tier 2 (country). *Fuse the two cities using the lifted operation*

$$\mathbf{Country} \;:=\; \Phi^{\uparrow}_{\text{link}}\big(\mathsf{U}_{\Sigma}(\mathbf{City}_A),\, \mathsf{U}_{\Sigma}(\mathbf{City}_B)\big).$$

This produces a tagged disjoint union of the two city carriers and adds bidirectional inter-city edges between every terminal of $\mathbf{City}_A$ *and every terminal of* $\mathbf{City}_B$*.*

Tier 3 (region). *Given several countries* $\mathbf{Country}_1, \ldots, \mathbf{Country}_r$*, form a region by*

$$\mathbf{Region} \;:=\; \Phi^{\uparrow}_{\text{link}}\big(\mathsf{U}_{\Sigma}(\mathbf{Country}_1), \ldots, \mathsf{U}_{\Sigma}(\mathbf{Country}_r)\big),$$

again adding links between country-level terminals. The result is a 3-level federation whose carrier is a multiply tagged union of stops, and whose relations are produced by the same *uniform recipe at each tier. This realizes an* Iterated MetaStructure of depth $t = 3$*: the tier-independent constructor* (Φ_{link}) *is lifted systematically by* U_{Σ} *to operate on structures-of-structures.*

2. Review and Result: HyperMatrix and Superhypermatrix

Matrices are rectangular arrays indexed by rows and columns and taking values in a ground algebra (typically a field) [38–44]. A *hypermatrix* extends this idea by allowing each entry to be a *set* of scalars rather than a single scalar [45]; iterating the powerset construction then yields *superhypermatrix* models that encode hierarchical uncertainty or multi–way choice (cf.[46]).

Definition 16 (Matrix)**.** *[47,48] Let K be a field (or skewfield) and let $I = \{1, \ldots, m\}$, $J = \{1, \ldots, n\}$. An* $m \times n$ matrix over K *is a function*

$$M : I \times J \longrightarrow K, \qquad (i, j) \longmapsto M(i, j) =: M_{ij}.$$

Addition and scalar multiplication are defined pointwise:

$$(M + N)_{ij} = M_{ij} + N_{ij}, \qquad (\lambda M)_{ij} = \lambda\, M_{ij} \quad (\lambda \in K).$$

Definition 17 (Hypermatrix (set–valued matrix))**.** *A* (set–valued) hypermatrix *over K is a map*

$$\mathcal{M} : I \times J \longrightarrow \mathcal{P}(K), \qquad (i, j) \longmapsto \mathcal{M}_{ij} \subseteq K,$$

where $\mathcal{P}(K)$ denotes the powerset of K. We extend linear operations entrywise via Minkowski lifting: for $A, B \subseteq K$ and $\lambda \in K$,

$$A \oplus B \;:=\; \{a + b : \; a \in A,\; b \in B\}, \qquad \lambda \odot A \;:=\; \{\lambda a : \; a \in A\}.$$

Thus, for hypermatrices $\mathcal{M}, \mathcal{N}$ and $\lambda \in K$,

$$(\mathcal{M} \oplus \mathcal{N})_{ij} := \mathcal{M}_{ij} \oplus \mathcal{N}_{ij}, \qquad (\lambda \odot \mathcal{M})_{ij} := \lambda \odot \mathcal{M}_{ij}.$$

The embedding $K \hookrightarrow \mathcal{P}(K)$, $a \mapsto \{a\}$, identifies every classical matrix M with the hypermatrix $\widehat{M}$ given by $\widehat{M}_{ij} = \{M_{ij}\}$.

Example 10 (Concrete 2×2 hypermatrix over $\mathbb{R}$)**.** *Let*

$$\mathcal{M} = \begin{bmatrix} [0,1] & \{2,3\} \\ \{0\} & \{1\} \end{bmatrix},$$

$$\mathcal{N} = \begin{bmatrix} \{1\} & [-1,1] \\ \{2\} & \{0\} \end{bmatrix},$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

where $[a,b] = \{x \in \mathbb{R} : a \le x \le b\}$*. Then*

$$\mathcal{M} \oplus \mathcal{N} = \begin{bmatrix} [1,2] & \{2,3\} \oplus [-1,1] \\ \{2\} & \{1\} \end{bmatrix} = \begin{bmatrix} [1,2] & [1,4] \\ \{2\} & \{1\} \end{bmatrix}, \quad 2 \odot \mathcal{M} = \begin{bmatrix} [0,2] & \{4,6\} \\ \{0\} & \{2\} \end{bmatrix}.$$

Definition 18 (n–Superhypermatrix and recursive lifts)**.** *For* $n \ge 1$*, let* $\mathcal{P}^1(K) = \mathcal{P}(K)$ *and* $\mathcal{P}^{n+1}(K) = \mathcal{P}(\mathcal{P}^n(K))$*. An* n–superhypermatrix *over* K *is a map*

$$\mathcal{M}^{\langle n \rangle} : I \times J \longrightarrow \mathcal{P}^n(K).$$

Define addition and scalar multiplication on $\mathcal{P}^n(K)$ *by recursion: for* $X, Y \in \mathcal{P}^n(K)$ *and* $\lambda \in K$*,*

$$X \boxplus^{(1)} Y := X \oplus Y, \quad \lambda \circledast^{(1)} X := \lambda \odot X;$$

$$X \boxplus^{(n+1)} Y := \{\, U \boxplus^{(n)} V : \ U \in X,\ V \in Y \,\}, \quad \lambda \circledast^{(n+1)} X := \{\, \lambda \circledast^{(n)} U : \ U \in X \,\}.$$

Operations on n*–superhypermatrices are then taken entrywise:*

$$\big(\mathcal{M}^{\langle n \rangle} \boxplus \mathcal{N}^{\langle n \rangle}\big)_{ij} := \mathcal{M}^{\langle n \rangle}_{ij} \boxplus^{(n)} \mathcal{N}^{\langle n \rangle}_{ij}, \qquad \big(\lambda \circledast \mathcal{M}^{\langle n \rangle}\big)_{ij} := \lambda \circledast^{(n)} \mathcal{M}^{\langle n \rangle}_{ij}.$$

The canonical embedding $\iota_n : K \to \mathcal{P}^n(K)$ *is given by iterated singletons:* $\iota_1(a) = \{a\}$ *and* $\iota_{n+1}(a) = \{\iota_n(a)\}$*, so that a classical matrix* M *embeds as* $\iota_n(M)_{ij} = \iota_n(M_{ij})$*.*

Example 11 (2–superhypermatrix: families of intervals)**.** *Let each entry be a finite* set of intervals*, i.e. an element of* $\mathcal{P}^2(\mathbb{R})$*:*

$$\mathcal{A}^{\langle 2 \rangle} = \begin{bmatrix} \{[0,1],\ [2,2]\} & \{[1,3]\} \\ \{[0,0]\} & \{[1,2],\ [-1,0]\} \end{bmatrix}, \qquad \mathcal{B}^{\langle 2 \rangle} = \begin{bmatrix} \{[1,1]\} & \{[-2,0],\ [0,0]\} \\ \{[2,3]\} & \{[0,0]\} \end{bmatrix}.$$

Their sum uses the level–2 rule $\boxplus^{(2)} = \{I \oplus J : \ I \in \cdot,\ J \in \cdot\}$ *with interval Minkowski sum:*

$$\big(\mathcal{A}^{\langle 2 \rangle} \boxplus \mathcal{B}^{\langle 2 \rangle}\big)_{11} = \{[0,1] \oplus [1,1],\ [2,2] \oplus [1,1]\} = \{[1,2],\ [3,3]\},$$

$$\big(\mathcal{A}^{\langle 2 \rangle} \boxplus \mathcal{B}^{\langle 2 \rangle}\big)_{12} = \{[1,3] \oplus [-2,0],\ [1,3] \oplus [0,0]\} = \{[-1,3],\ [1,3]\},$$

and similarly for the other entries. Scalar multiplication (e.g. $2 \circledast \mathcal{A}^{\langle 2 \rangle}$*) doubles every interval in every set at level* 2*.*

Notation 2. *Fix a (skew)field* K*, finite index sets* $I = \{1, \dots, p\}$ *and* $J = \{1, \dots, q\}$*, and integers* $1 \le m \le n$*. Write* $\mathcal{P}^1(K) = \mathcal{P}(K)$ *and* $\mathcal{P}^{r+1}(K) = \mathcal{P}(\mathcal{P}^r(K))$ *for* $r \ge 1$*. We use two canonical maps between levels:*

$$\iota_{r \to s} : \mathcal{P}^r(K) \longrightarrow \mathcal{P}^s(K) \quad (s \ge r), \qquad \iota_{r \to r} = \mathrm{id}, \ \ \iota_{r \to s}(X) = \{\iota_{r \to s-1}(X)\}$$

(nested singleton lift), and the level–lowering maps

$$\mu_{s \to r} : \mathcal{P}^s(K) \longrightarrow \mathcal{P}^r(K) \quad (s \ge r \ge 1), \qquad \mu_{t \to t-1}(X) = \bigcup X, \ \ \mu_{s \to r} = \mu_{r+1 \to r} \circ \cdots \circ \mu_{s \to s-1}.$$

Definition 19 (Recursive m–level lift of scalar operations)**.** *Let* $\oplus, \otimes : K \times K \to K$ *be the field addition and multiplication, and for* $\lambda \in K$ *let* $\lambda \cdot (\,\cdot\,) : K \to K$ *be scalar multiplication. Define their* m–level set lifts $\boxplus^{(m)}, \boxtimes^{(m)} : \mathcal{P}^m(K) \times \mathcal{P}^m(K) \to \mathcal{P}^m(K)$ *and* $\circledast^{(m)} : K \times \mathcal{P}^m(K) \to \mathcal{P}^m(K)$ *recursively by*

$$A \boxplus^{(1)} B = \{a \oplus b : a \in A,\ b \in B\}, \quad A \boxtimes^{(1)} B = \{a \otimes b : a \in A,\ b \in B\}, \quad \lambda \circledast^{(1)} A = \{\lambda \cdot a : \ a \in A\},$$

$$X \boxplus^{(r+1)} Y = \{U \boxplus^{(r)} V : U \in X,\ V \in Y\}, \quad X \boxtimes^{(r+1)} Y = \{U \boxtimes^{(r)} V : U \in X,\ V \in Y\},$$

$$\lambda \circledast^{(r+1)} X = \{\lambda \circledast^{(r)} U : U \in X\}.$$

Definition 20 ((m,n)–lifted operations on level n)**.** *For $X, Y \in \mathcal{P}^n(K)$ and $\lambda \in K$ define*

$$X \boxplus^{(m,n)} Y := \iota_{m\to n}\Big(\mu_{n\to m}(X) \boxplus^{(m)} \mu_{n\to m}(Y) \Big),$$

$$X \boxtimes^{(m,n)} Y := \iota_{m\to n}\Big(\mu_{n\to m}(X) \boxtimes^{(m)} \mu_{n\to m}(Y) \Big),$$

$$\lambda \circledast^{(m,n)} X := \iota_{m\to n}\Big(\lambda \circledast^{(m)} \mu_{n\to m}(X) \Big).$$

Thus we flatten *inputs from level n down to level m, perform the m–level Minkowski–type operation, and* re–lift *to level n.*

Definition 21 ((m,n)–Superhypermatrix)**.** *A (m,n)–superhypermatrix over K with shape $p \times q$ is a function*

$$\mathcal{M} : I \times J \longrightarrow \mathcal{P}^n(K).$$

We define addition and scalar multiplication entrywise via Definition 20:

$$(\mathcal{M} \boxplus \mathcal{N})_{ij} := \mathcal{M}_{ij} \boxplus^{(m,n)} \mathcal{N}_{ij}, \qquad (\lambda \circledast \mathcal{M})_{ij} := \lambda \circledast^{(m,n)} \mathcal{M}_{ij}.$$

*If J is finite, the (m,n)–*matrix product *of $\mathcal{M} \in \mathcal{P}^n(K)^{I\times J}$ and $\mathcal{N} \in \mathcal{P}^n(K)^{J\times L}$ is defined by*

$$(\mathcal{M} \boxtimes \mathcal{N})_{il} := \boxplus^{(m,n)}_{j\in J} \Big(\mathcal{M}_{ij} \boxtimes^{(m,n)} \mathcal{N}_{jl} \Big),$$

where $\boxplus^{(m,n)}$ denotes the iterated (m,n)–sum (well–defined since J is finite).

Remark 2 (Selections and realizations)**.** *A* selection *of $\mathcal{M}$ chooses, for every (i,j), an element of $\mu_{n\to 1}(\mathcal{M}_{ij}) \in \mathcal{P}(K)$ and then an element of K, yielding a classical matrix $M \in K^{I\times J}$. Hence every (m,n)–superhypermatrix encodes a (possibly large) family of ordinary matrices compatible with its entries.*

Example 12 (A concrete $(m,n) = (1,2)$ superhypermatrix)**.** *Let $K = \mathbb{R}$, $I = J = \{1,2\}$, $m = 1$, $n = 2$, and*

$$\mathcal{M}_{11} = \{[0,1]\},\ \mathcal{M}_{12} = \{[1,2],[3,3]\},\ \mathcal{M}_{21} = \{\{0\}\},\ \mathcal{M}_{22} = \{[-1,0]\} \subseteq \mathcal{P}^2(\mathbb{R}),$$

where $[a,b] = \{x \in \mathbb{R} : a \le x \le b\}$ and $\{0\} \in \mathcal{P}^1(\mathbb{R})$ is lifted to level 2 by a singleton brace. For addition, flatten to level 1 via $\mu_{2\to 1}$ (a union of the displayed sets of intervals), apply $\boxplus^{(1)}$ (interval Minkowski sum or setwise sum), then re–lift by $\iota_{1\to 2}$. Scalar multiplication uses the same pattern with $\circledast^{(1)}$ (interval scaling). Thus the $(1,2)$–rules combine familiar interval arithmetic at level 1 with a level–2 wrapper that preserves hierarchical structure.

Example 13 ((m,n)=(1,2) — Delivery time planning with route alternatives)**.** *Fix $K = \mathbb{R}_{\ge 0}$ (minutes). Each matrix entry is a* finite set of intervals *(an element of $\mathcal{P}^2(K)$): each interval $[a,b]$ is a plausible time window for one route option, and a set of intervals collects the* route alternatives *available in that cell. Let rows be* carriers *$I = \{C_1, C_2\}$ and columns be* legs *$J = \{L_1, L_2\}$. Consider*

$$\mathcal{T} = \begin{bmatrix} \{[30,35],\ [40,45]\} & \{[20,25]\} \\ \{[35,40]\} & \{[15,20],\ [18,22]\} \end{bmatrix} \in \mathcal{P}^2(K)^{I\times J}.$$

Let the leg–count vector (how many times each leg is taken) be

$$\rfloor = \begin{bmatrix} \{\{1\}\} \\ \{\{1\}\} \end{bmatrix} \in \mathcal{P}^2(K)^{J\times\{1\}},$$

i.e., one unit of L_1 *and one unit of* L_2*. With* $(m,n)=(1,2)$*, the matrix product*

$$(\mathcal{T}\boxtimes \rfloor)_{i1}=\boxplus^{(1,2)}_{j\in J}\left(\mathcal{T}_{ij}\boxtimes^{(1,2)}\rfloor_{j1}\right)\in\mathcal{P}^2(K)$$

first flattens to level 1 *(sets of scalars), then performs the usual (level–1) Minkowski product/sum, and finally re–lifts to level 2.*

Carrier C_1**.**

$$\mathcal{T}_{11}\boxtimes^{(1,2)}\rfloor_{11}=\{[30,35],[40,45]\}\boxtimes^{(1,2)}\{\{1\}\}=\{[30,35],[40,45]\},$$

$$\mathcal{T}_{12}\boxtimes^{(1,2)}\rfloor_{21}=\{[20,25]\}\boxtimes^{(1,2)}\{\{1\}\}=\{[20,25]\}.$$

Summing legs by $\boxplus^{(1,2)}$*:*

$$(\mathcal{T}\boxtimes\rfloor)_{11}=\{[30,35],[40,45]\}\boxplus^{(1,2)}\{[20,25]\}=\{[30{+}20,\ 35{+}25],\ [40{+}20,\ 45{+}25]\}=\{[50,60],\ [60,70]\}.$$

Carrier C_2**.**

$$\mathcal{T}_{21}\boxtimes^{(1,2)}\rfloor_{11}=\{[35,40]\},\qquad \mathcal{T}_{22}\boxtimes^{(1,2)}\rfloor_{21}=\{[15,20],[18,22]\},$$

$$(\mathcal{T}\boxtimes\rfloor)_{21}=\{[35,40]\}\ \boxplus^{(1,2)}\ \{[15,20],[18,22]\}=\{[50,60],\ [53,62]\}.$$

For C_1 *the total delivery time is either* $[50,60]$ *or* $[60,70]$ *minutes, depending on the route combination; for* C_2 *it is either* $[50,60]$ *or* $[53,62]$ *minutes. The* $(1,2)$*–superhypermatrix keeps the* family of feasible totals *rather than a single number.*

Example 14 **((m,n)=(1,2) — Procurement with uncertain quotes and quantities).** *Let* $K=\mathbb{R}_{\geq 0}$ *(USD). Rows are* vendors $I=\{\mathrm{A},\mathrm{B}\}$ *and columns are* parts $J=\{\mathrm{P}_1,\mathrm{P}_2\}$*. Each entry is a* set of unit–price intervals *capturing promotional/market uncertainty:*

$$\mathcal{Q}\ =\ \begin{bmatrix}\{[9,11]\} & \{[14,16],\ [13,17]\}\\ \{[8,10],\ [9,12]\} & \{[15,18]\}\end{bmatrix}\ \in\ \mathcal{P}^2(K)^{I\times J}.$$

Quantities (as a column) are exact integers, encoded at level 2 by singleton lifts:

$$\mathcal{N}\ =\ \begin{bmatrix}\{\{10\}\}\\ \{\{5\}\}\end{bmatrix}\ \in\ \mathcal{P}^2(K)^{J\times\{1\}}.$$

The total spend per vendor is $(\mathcal{Q}\boxtimes\mathcal{N})_{i1}\in\mathcal{P}^2(K)$*.*

Vendor **A.** *Compute the two leg terms:*

$$\mathcal{Q}_{\mathrm{A},\mathrm{P}_1}\boxtimes^{(1,2)}\mathcal{N}_{\mathrm{P}_1,1}=\{[9,11]\}\boxtimes^{(1,2)}\{\{10\}\}=\{[90,110]\},$$

$$\mathcal{Q}_{\mathrm{A},\mathrm{P}_2}\boxtimes^{(1,2)}\mathcal{N}_{\mathrm{P}_2,1}=\{[14,16],[13,17]\}\boxtimes^{(1,2)}\{\{5\}\}=\{[70,80],\ [65,85]\}.$$

Sum by $\boxplus^{(1,2)}$*:*

$$(\mathcal{Q}\boxtimes\mathcal{N})_{\mathrm{A},1}=\{[90,110]\}\ \boxplus^{(1,2)}\ \{[70,80],[65,85]\}=\{\,[160,190],\ [155,195]\,\}.$$

Thus Vendor A's total is either the interval $[160,190]$ *(if* P_2 *clears at* $[14,16]$*) or* $[155,195]$ *(if it clears at* $[13,17]$*).*

Vendor **B.**

$$\mathcal{Q}_{\mathrm{B},\mathrm{P}_1}\boxtimes^{(1,2)}\mathcal{N}_{\mathrm{P}_1,1}=\{[8,10],[9,12]\}\boxtimes^{(1,2)}\{\{10\}\}=\{[80,100],\ [90,120]\},$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

$$\mathcal{Q}_{\mathrm{B,P_2}} \boxtimes^{(1,2)} \mathcal{N}_{\mathrm{P_2},1} = \{[15,18]\} \boxtimes^{(1,2)} \{\{5\}\} = \{[75,90]\}.$$

Hence

$$(\mathcal{Q} \boxtimes \mathcal{N})_{\mathrm{B},1} = \{[80,100],[90,120]\} \boxplus^{(1,2)} \{[75,90]\} = \{\,[155,190],\ [165,210]\,\}.$$

The $(1,2)$*–superhypermatrix product returns a* family of plausible order totals *per vendor, explicitly propagating interval uncertainty (quotes) through multiplication by exact quantities and aggregation across parts.*

Theorem 1 (Reduction to n–superhypermatrix). *Fix $n \geq 1$ and take $m = n$. Then the operations $\boxplus^{(n,n)}$, $\boxtimes^{(n,n)}$, and $\circledast^{(n,n)}$ coincide with the standard level–n recursive lifts $\boxplus^{(n)}$, $\boxtimes^{(n)}$, and $\circledast^{(n)}$ from Definition 19. Consequently, every n–superhypermatrix (i.e. a map $I \times J \to \mathcal{P}^n(K)$ with entrywise level–n operations) is a special case of an (m,n)–superhypermatrix (namely with $m = n$).*

Proof. By Definition 20, when $m = n$ we have $\mu_{n\to n} = \mathrm{id}$ and $\iota_{n\to n} = \mathrm{id}$, hence

$$X \boxplus^{(n,n)} Y = \iota_{n\to n}\big(\mu_{n\to n}(X) \boxplus^{(n)} \mu_{n\to n}(Y)\big) = X \boxplus^{(n)} Y,$$

and similarly for $\boxtimes$ and $\circledast$. Therefore the (m,n)–entrywise operations reduce to the usual level–n recursive Minkowski lifts, proving the claim. □

Theorem 2 (Well–definedness and closure). *Let $1 \leq m \leq n$ and $\mathcal{M}, \mathcal{N} \in \mathcal{P}^n(K)^{I\times J}$. Then $\mathcal{M} \boxplus \mathcal{N}$ and $\lambda \circledast \mathcal{M}$ (for any $\lambda \in K$) are again in $\mathcal{P}^n(K)^{I\times J}$. If J is finite and $\mathcal{N} \in \mathcal{P}^n(K)^{J\times L}$, then $\mathcal{M} \boxtimes \mathcal{N} \in \mathcal{P}^n(K)^{I\times L}$.*

Proof. By construction $\mu_{n\to m}(X) \in \mathcal{P}^m(K)$ for every $X \in \mathcal{P}^n(K)$. Definition 19 yields $\mu_{n\to m}(X) \boxplus^{(m)} \mu_{n\to m}(Y) \in \mathcal{P}^m(K)$ and $\lambda \circledast^{(m)} \mu_{n\to m}(X) \in \mathcal{P}^m(K)$. Applying $\iota_{m\to n}$ returns elements of $\mathcal{P}^n(K)$, proving entrywise closure for $\boxplus$ and $\circledast$. For products, each $\mathcal{M}_{ij} \boxtimes^{(m,n)} \mathcal{N}_{jl} \in \mathcal{P}^n(K)$ and a finite iterated $\boxplus^{(m,n)}$ remains in $\mathcal{P}^n(K)$. □

3. Review and Result: MultiMatrix and Iterative Multimatrix

A MultiMatrix is a matrix whose entries are finite multisets of scalars, with operations lifted entrywise via multiset Minkowski rules. An Iterative Multimatrix stacks MultiMatrices across levels, each entry a multiset-of-multisets, combining levelwise via lifted operations to model hierarchical aggregation.

Notation 3. *For a nonempty set H, a (finite) multiset on H is a function $m : H \to \mathbb{N}_0$ with finite support. The collection of all finite multisets on H is denoted $\mathcal{M}(H)$. For $A, B \in \mathcal{M}(H)$ and $h \in H$, write $A(h)$ for the multiplicity of h. Define the* multiset Minkowski lifts *of a binary map $\star : H \times H \to H$ and of a unary map $u : H \to H$ by*

$$(A \mathbin{\widehat{\star}} B)(t) \;:=\; \sum_{\substack{a,b\in H\\ a\star b=t}} A(a)\,B(b), \qquad (\widehat{u}(A))(t) \;:=\; \sum_{\substack{a\in H\\ u(a)=t}} A(a).$$

When H is a (skew)field K with $+$ and $\cdot$, we abbreviate

$$A \boxplus B \;:=\; A \mathbin{\widehat{+}} B, \qquad A \boxtimes B \;:=\; A \mathbin{\widehat{\cdot}} B, \qquad \lambda \circledast A \;:=\; \widehat{(a \mapsto \lambda a)}(A).$$

Notation 4 (Indexing). *Fix finite index sets $I = \{1,\ldots,p\}$ and $J = \{1,\ldots,q\}$, and set $X := I \times J$.*

Definition 22 (MultiMatrix over a field). *Let K be a (skew)field. A* MultiMatrix *of shape $p \times q$ over K is a function*

$$\mathcal{A} : X = I \times J \longrightarrow \mathcal{M}(K), \qquad (i,j) \mapsto \mathcal{A}_{ij},$$

 doi:10.20944/preprints202510.0232.v1

i.e., each entry is a finite multiset of scalars. Define operations entrywise by

$$(\mathcal{A} \oplus \mathcal{B})_{ij} := \mathcal{A}_{ij} \boxplus \mathcal{B}_{ij}, \qquad (\lambda \odot \mathcal{A})_{ij} := \lambda \circledast \mathcal{A}_{ij},$$

and, when J is finite, the MultiMatrix product *by*

$$(\mathcal{A} \odot \mathcal{B})_{il} := \boxplus_{j \in J} \left(\mathcal{A}_{ij} \boxtimes \mathcal{B}_{jl} \right), \qquad \mathcal{A} \in \mathcal{M}(K)^{I \times J},\ \mathcal{B} \in \mathcal{M}(K)^{J \times L},$$

where $\boxplus$ denotes finite iteration of $\boxplus$.

Example 15 (MultiMatrix — weighted course grading (two components, two students))**.** *Let $K = \mathbb{R}$, $I = J = \{1,2\}$, and write multisets with multiplicities as $\{\{x_1^{k_1}, \ldots, x_r^{k_r}\}\}$. Consider the* score MultiMatrix

$$\mathcal{S} = \begin{bmatrix} \{\{78, 80\}\} & \{\{88, 90\}\} \\ \{\{70\}\} & \{\{92, 93^2\}\} \end{bmatrix} \in \mathcal{M}(\mathbb{R})^{I \times J},$$

where each entry lists all available scores for a student–component pair (e.g., multiple graders or attempts; the entry 93^2 means two identical 93's).

Let the (column) weight MultiMatrix *be*

$$\mathcal{W} = \begin{bmatrix} \{\{0.4\}\} \\ \{\{0.6\}\} \end{bmatrix} \in \mathcal{M}(\mathbb{R})^{J \times \{1\}},$$

so component 1 carries weight 0.4 and component 2 weight 0.6. Using the MultiMatrix product from Definition 22,

$$(\mathcal{S} \odot \mathcal{W})_{i1} = \boxplus_{j=1}^{2} \left(\mathcal{S}_{ij} \boxtimes \mathcal{W}_{j1} \right),$$

where $\boxtimes$ (resp. $\boxplus$) is the multiset lift of scalar multiplication (resp. addition), we obtain a multiset of weighted totals *for each student.*

Student 1.

$$\mathcal{S}_{11} \boxtimes \mathcal{W}_{11} = \{\{0.4\cdot 78,\ 0.4\cdot 80\}\} = \{\{31.2,\ 32\}\}, \qquad \mathcal{S}_{12} \boxtimes \mathcal{W}_{21} = \{\{0.6\cdot 88,\ 0.6\cdot 90\}\} = \{\{52.8,\ 54\}\}.$$

Thus

$$(\mathcal{S} \odot \mathcal{W})_{11} = \{\{31.2,\ 32\}\} \boxplus \{\{52.8,\ 54\}\} = \{\{84.0,\ 85.2,\ 84.8,\ 86.0\}\}.$$

Student 2.

$$\mathcal{S}_{21} \boxtimes \mathcal{W}_{11} = \{\{28\}\}, \qquad \mathcal{S}_{22} \boxtimes \mathcal{W}_{21} = \{\{55.2,\ 55.8^2\}\},$$

hence

$$(\mathcal{S} \odot \mathcal{W})_{21} = \{\{28\}\} \boxplus \{\{55.2,\ 55.8^2\}\} = \{\{83.2^1,\ 83.8^2\}\}.$$

Therefore $\mathcal{S} \odot \mathcal{W} \in \mathcal{M}(\mathbb{R})^{I \times \{1\}}$ returns, for each student, the finite multiset of all possible weighted course totals, *capturing grading variability (multiple graders/attempts) while remaining compatible with standard matrix weighting when entries are singletons.*

Lemma 1 (Closure)**.** *For MultiMatrices $\mathcal{A}, \mathcal{B}$ of the same shape and $\lambda \in K$, $\mathcal{A} \oplus \mathcal{B}$ and $\lambda \odot \mathcal{A}$ are MultiMatrices of that shape. If $\mathcal{A} \in \mathcal{M}(K)^{I \times J}$ and $\mathcal{B} \in \mathcal{M}(K)^{J \times L}$, then $\mathcal{A} \odot \mathcal{B} \in \mathcal{M}(K)^{I \times L}$.*

Proof. Each operation is built from $\boxplus, \boxtimes, \circledast$ on $\mathcal{M}(K)$, which are closed by construction of the lifted multiplicities. Finite iteration of $\boxplus$ remains in $\mathcal{M}(K)$, establishing entrywise closure. □

 doi:10.20944/preprints202510.0232.v1

Theorem 3 (MultiMatrix as a MultiStructure and reduction to classical matrices)**.** *Let $H := K$ and define multi-operations on H by*

$$\#_{+}^{(2)}(a,b) := \{\{a+b\}\}, \qquad \#_{\cdot}^{(2)}(a,b) := \{\{ab\}\}, \qquad \#_{\lambda}^{(1)}(a) := \{\{\lambda a\}\}.$$

Then the pair $\mathcal{MS}_K = (H, \{\#_{+}^{(2)}, \#_{\cdot}^{(2)}, \#_{\lambda}^{(1)}\}_{\lambda\in K})$ is a MultiStructure *(maps from tuples to finite multisets). Its pointwise (dimensional) lift along the axis set $X = I \times J$,*

$$\mathcal{MS}_K^{[X]} : (H^X)^m \longrightarrow (\mathcal{M}(H))^X, \quad (f_1,\ldots,f_m) \mapsto [d \mapsto \#^{(m)}(f_1(d),\ldots,f_m(d))],$$

has codomain exactly the set of MultiMatrices $\mathcal{M}(K)^X$. Moreover, if we embed classical matrices $M \in K^X$ by the singleton lift $\sigma(M)_{ij} := \{\{M_{ij}\}\}$*, then*

$$\sigma(M+N) = \sigma(M) \oplus \sigma(N), \qquad \sigma(\lambda M) = \lambda \odot \sigma(M), \qquad \sigma(MN) = \sigma(M) \odot \sigma(N).$$

Hence MultiMatrix generalizes classical matrix algebra (recovering it on singleton entries) and is representable via the MultiStructure lift $\mathcal{MS}_K^{[X]}$.

Proof. The multi-operations $\#^{(m)}$ are finite-multiset valued by definition, so $\mathcal{MS}_K$ is a MultiStructure. The dimensional lift (Definition of pointwise lift) evaluates $\#^{(m)}$ at each index $d \in X$, producing an element of $(\mathcal{M}(H))^X$, i.e., a MultiMatrix. For a classical matrix M, $\sigma(M)$ has singleton entries. Because the lifted multiset operations reduce to classical operations on singletons, $\boxplus, \boxtimes, \circledast$ coincide with $+, \cdot, (\lambda\,\cdot)$, respectively, yielding the displayed equalities and the reduction. □

Remark 3 (Selections viewpoint)**.** *A MultiMatrix $\mathcal{A}$ induces a (finite) multiset of ordinary matrices: choose for each (i,j) an element a_{ij} from $\mathcal{A}_{ij}$ (counted with product of multiplicities). Under this viewpoint, $\oplus$ (resp. $\odot$) corresponds to the multiset sum of pairwise matrix sums (resp. products), consistent with Definition 22.*

Definition 23 (Iterative MultiMatrix of depth k)**.** *Fix $k \in \mathbb{N}$. An* Iterative MultiMatrix (IMM) of depth k *and shape $p \times q$ is a tuple*

$$\boldsymbol{\mathcal{A}} = \left(\mathcal{A}^{(0)}, \mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)} \right), \qquad \mathcal{A}^{(r)} \in \left(\mathcal{M}^r(K)\right)^{I\times J}.$$

Operations act levelwise and entrywise *using the lifted maps on $\mathcal{M}^r(K)$:*

$$(\boldsymbol{\mathcal{A}} \oplus \boldsymbol{\mathcal{B}})_{ij}^{(r)} := \mathcal{A}_{ij}^{(r)} \mathbin{\widehat{+}} \mathcal{B}_{ij}^{(r)}, \qquad (\lambda \odot \boldsymbol{\mathcal{A}})_{ij}^{(r)} := \widehat{(a \mapsto \lambda a)}(\mathcal{A}_{ij}^{(r)}),$$

and, for multiplication when J is finite,

$$(\boldsymbol{\mathcal{A}} \odot \boldsymbol{\mathcal{B}})_{il}^{(r)} := \widehat{\sum_{j\in J}} \left(\mathcal{A}_{ij}^{(r)} \mathbin{\widehat{\cdot}} \mathcal{B}_{jl}^{(r)} \right) \quad \text{for each level } r = 0,\ldots,k.$$

Example 16 (Iterative MultiMatrix — warehouse packing across levels (items → orders → truckloads))**.** *Let the ground field be $K = \mathbb{R}_{\geq 0}$ (weights in kg). Take two orders $I = \{O_1, O_2\}$ and two delivery windows $J = \{D_1, D_2\}$. We build an* Iterative MultiMatrix *of depth $k = 2$, $\boldsymbol{\mathcal{A}} = (\mathcal{A}^{(0)}, \mathcal{A}^{(1)}, \mathcal{A}^{(2)})$, where $\mathcal{A}^{(r)} \in \left(\mathcal{M}^r(K)\right)^{I\times J}$.*

Level 0 (single pallet option per cell).

$\mathcal{A}^{(0)} =$ (kg)

	D_1	D_2
O_1	240	220
O_2	120	150

Each entry is one representative pallet weight for that (order, window).

Level 1 (multiset of pallets per cell). *Here $\mathcal{A}^{(1)}_{ij} \in \mathcal{M}(K)$ collects all pallets planned for that cell (multiplicity encodes how many identical pallets). We write multisets as $\{\{x_1^{k_1}, \ldots, x_r^{k_r}\}\}$.*

$$\mathcal{A}^{(1)} = \begin{array}{c|cc} & \mathrm{D}_1 & \mathrm{D}_2 \\ \hline \mathrm{O}_1 & \{\{240^2\}\} & \{\{220^1, 180^1\}\} \\ \mathrm{O}_2 & \{\{120^3\}\} & \{\{150^1, 130^1\}\} \end{array}$$

Example: $\mathcal{A}^{(1)}_{\mathrm{O}_1,\mathrm{D}_1} = \{\{240, 240\}\}$ means two identical pallets of 240 kg.

Level 2 (multiset of **load plans** ***per cell).*** *Now $\mathcal{A}^{(2)}_{ij} \in \mathcal{M}(\mathcal{M}(K))$; each element of $\mathcal{A}^{(2)}_{ij}$ is itself a multiset of pallets (i.e., one admissible truckload composition for that cell). For readability, we list the four cells separately:*

$$\mathcal{A}^{(2)}_{\mathrm{O}_1,\mathrm{D}_1} = \{\{\ \underbrace{\{\{240^2\}\}}_{\textit{plan A}},\ \underbrace{\{\{180^2, 100^1\}\}}_{\textit{plan B}}\ \}\},$$

$$\mathcal{A}^{(2)}_{\mathrm{O}_1,\mathrm{D}_2} = \{\{\ \underbrace{\{\{220^1, 180^1\}\}}_{\textit{plan C}},\ \underbrace{\{\{200^2\}\}}_{\textit{plan D}}\ \}\},$$

$$\mathcal{A}^{(2)}_{\mathrm{O}_2,\mathrm{D}_1} = \{\{\ \underbrace{\{\{120^3\}\}}_{\textit{plan E}}\ \}\},$$

$$\mathcal{A}^{(2)}_{\mathrm{O}_2,\mathrm{D}_2} = \{\{\ \underbrace{\{\{150^1, 130^1\}\}}_{\textit{plan F}},\ \underbrace{\{\{140^2\}\}}_{\textit{plan G}}\ \}\}.$$

Interpretation: for $(\mathrm{O}_1, \mathrm{D}_1)$ there are two feasible truckload plans: either two pallets of 240 kg (plan A) or an alternative mix (two pallets of 180 kg plus one of 100 kg, plan B). Thus the level–2 entry is a multiset of admissible pallet-multisets.

How levelwise operations act. *Given another IMM $\boldsymbol{\mathcal{B}}$ (e.g., a second warehouse), the sum $(\boldsymbol{\mathcal{A}} \oplus \boldsymbol{\mathcal{B}})^{(0)}$ adds scalar weights cellwise; $(\cdot)^{(1)}$ performs the multiset lift of addition on pallet multisets; and $(\cdot)^{(2)}$ combines load plans by the lifted rule on $\mathcal{M}(\mathcal{M}(K))$ (Definition 23). This preserves the three-tier meaning: single pallets $\to$ pallet collections $\to$ sets of admissible load plans.*

Lemma 2 (Closure per level)**.** *For each r, the lifted maps $\widehat{+}, \widehat{\cdot}$ and $\widehat{(a \mapsto \lambda a)}$ send $\left(\mathcal{M}^r(K)\right)^{I \times J}$ to itself, hence the IMM operations are well defined.*

Proof. Identical to Lemma 1, applied in the universe $\mathcal{M}^r(K)$. □

Theorem 4 (Iterative MultiMatrix as an Iterative MultiStructure)**.** *For each $k \geq 0$, set $H := K$ and define levelwise multi-operations*

$$\#^{(m,r)} : \left(\mathcal{M}^r(H)\right)^m \longrightarrow \mathcal{M}^r(H), \qquad \#^{(2,r)}_{+} = \widehat{+},\ \#^{(2,r)}_{\cdot} = \widehat{\cdot},\ \#^{(1,r)}_{\lambda} = \widehat{(a \mapsto \lambda a)}.$$

Then

$$\mathcal{IMS}^{(k)}_K := \left(H,\ \{\#^{(m,r)}\}_{m \in \{1,2\},\ 0 \leq r \leq k}\right)$$

is an Iterative MultiStructure *in the sense of level-indexed multi-operations. Moreover, the dimensional lift of $\mathcal{IMS}^{(k)}_K$ along $X = I \times J$, acting pointwise at each level r, yields exactly the IMM space of Definition 23.*

Proof. By construction, each $\#^{(m,r)}$ maps m-tuples in $\mathcal{M}^r(H)$ to an element of $\mathcal{M}^r(H)$, so the tuple forms an Iterative MultiStructure. Lifting along X replaces elements by X-indexed arrays and applies the same operations entrywise, which is precisely the IMM definition. □

Theorem 5 (Reductions: IMM $\Rightarrow$ MultiMatrix $\Rightarrow$ Matrix)**.** *1. For $k = 1$, an IMM is a single matrix with entries in $\mathcal{M}^1(K) = \mathcal{M}(K)$, with operations from Definition 22; hence IMM generalizes MultiMatrix.*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0232.v1

2. *For $k = 0$, an IMM is a classical matrix with entries in $\mathcal{M}^0(K) = K$, and the lifted operations reduce to standard matrix algebra; hence MultiMatrix generalizes classical matrices via the singleton embedding.*

Proof. (1) Immediate from the definitions with $r = 0, 1$.

(2) When $r = 0$, the lifts $\widehat{+}, \widehat{\cdot}$ coincide with $+, \cdot$ on K, and $\widehat{(a \mapsto \lambda a)}$ with scalar multiplication, so we recover ordinary matrix operations. The singleton embedding argument is as in Theorem 3. □

4. Review and Result: MetaMatrix and Iterated MetaMatrix

MetaMatrix is a matrix whose entries are matrices; operations act uniformly by lifting row–column arithmetic to block-level structural composition rules. Iterated MetaMatrix stacks MetaMatrices across levels, forming matrices of matrices of matrices, with operations defined recursively and naturally across depths.

Definition 24 (Block profile and admissibility)**.** *Let $I = \{1, \dots, p\}$ and $J = \{1, \dots, q\}$ be finite index sets. A* block profile *on (I, J) consists of two dimension vectors*

$$\mathbf{r} = (r_i)_{i \in I} \in (\mathbb{N}_{>0})^I, \qquad \mathbf{s} = (s_j)_{j \in J} \in (\mathbb{N}_{>0})^J.$$

We say that two profiles $(I, J; \mathbf{r}, \mathbf{s})$ and $(J, L; \mathbf{s}, \mathbf{t})$ are multiplication–compatible *if their shared inner dimension vector is identical ($\mathbf{s}$ on both), where $L = \{1, \dots, \ell\}$ and $\mathbf{t} = (t_\ell)_{\ell \in L}$.*

Definition 25 (MetaMatrix (matrix of matrices))**.** *Given a block profile $(I, J; \mathbf{r}, \mathbf{s})$, a* MetaMatrix over K *with that profile is a function*

$$\mathbb{A} \,:\, I \times J \longrightarrow \bigsqcup_{(i,j) \in I \times J} K^{r_i \times s_j}, \qquad (i, j) \longmapsto A_{ij} \in K^{r_i \times s_j}.$$

If $(I, J; \mathbf{r}, \mathbf{s}) = (I, J; \mathbf{r}', \mathbf{s}')$ we define blockwise addition *and* scalar multiplication *entrywise:*

$$(\mathbb{A} \oplus \mathbb{B})_{ij} \;:=\; A_{ij} + B_{ij}, \qquad (\lambda \odot \mathbb{A})_{ij} \;:=\; \lambda A_{ij}.$$

If $(I, J; \mathbf{r}, \mathbf{s})$ and $(J, L; \mathbf{s}, \mathbf{t})$ are multiplication–compatible, the MetaMatrix product *is*

$$(\mathbb{A} \,\otimes\, \mathbb{B})_{i\ell} \;:=\; \sum_{j \in J} A_{ij}\, B_{j\ell} \quad \in \; K^{r_i \times t_\ell}.$$

Example 17 (MetaMatrix — hospital staffing (roles × shifts within wards × days))**.** *Fix the ground field $K = \mathbb{N}$ (headcounts). Let rows index* wards $I = \{\mathrm{ICU}, \mathrm{GEN}\}$ *and columns index* days $J = \{\mathrm{Mon}, \mathrm{Tue}\}$*. Each block $A_{ij} \in K^{r_i \times s_j}$ is a 2×2 matrix whose rows are $\{\mathrm{RN}, \mathrm{LPN}\}$ (roles) and whose columns are $\{\mathrm{Day}, \mathrm{Night}\}$ (shifts). Thus the block profile is $(I, J; \mathbf{r}, \mathbf{s})$ with $\mathbf{r} = (2, 2)$, $\mathbf{s} = (2, 2)$.*

Define the MetaMatrix $\mathbb{S} : I \times J \to K^{2\times 2}$ by

$$\mathbb{S} \;=\; \begin{array}{c|cc} & \text{Mon} & \text{Tue} \\ \hline \text{ICU} & \begin{bmatrix} 5 & 4 \\ 2 & 2 \end{bmatrix} & \begin{bmatrix} 5 & 3 \\ 2 & 2 \end{bmatrix} \\ \text{GEN} & \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix} & \begin{bmatrix} 4 & 4 \\ 3 & 2 \end{bmatrix} \end{array} \quad \in \quad \prod_{(i,j) \in I \times J} K^{2\times 2}.$$

 doi:10.20944/preprints202510.0232.v1

Its flattening is the 4×4 *block–assembled matrix*

$$
[\![\mathbb{S}]\!] = \begin{array}{c} \\ \end{array}
\begin{array}{cccc} \text{Mon Day} & \text{Mon Night} & \text{Tue Day} & \text{Tue Night} \\ \end{array}
$$

$$
[\![\mathbb{S}]\!] = \begin{bmatrix} 5 & 4 & 5 & 3 \\ 2 & 2 & 2 & 2 \\ 4 & 3 & 4 & 4 \\ 3 & 2 & 3 & 2 \end{bmatrix}
$$

(rows ordered as $\mathrm{ICU}:\mathrm{RN}, \mathrm{ICU}:\mathrm{LPN}, \mathrm{GEN}:\mathrm{RN}, \mathrm{GEN}:\mathrm{LPN}$*).*

Aggregating across shifts. *Let* $\nVdash_{\text{shift}}$ *be the block column with profile* $(J, \{\star\}; \mathbf{s}, \mathbf{t})$ *where* $\mathbf{t} = (1)$ *and, for each* $j \in J$,

$$
(\nVdash_{\text{shift}})_{j\star} := \begin{bmatrix} 1 \\ 1 \end{bmatrix} \in K^{2\times 1}.
$$

The MetaMatrix product (Definition 25) gives a block column $(\mathbb{S} \otimes \nVdash_{\text{shift}})_{i\star} = \sum_{j\in J} A_{ij}\,[1 \quad 1]^\top \in K^{2\times 1}$*. Concretely,*

$$
(\mathbb{S} \otimes \nVdash_{\text{shift}})_{\mathrm{ICU},\star} = \begin{bmatrix} 5 & 4 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 5 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 9 \\ 4 \end{bmatrix} + \begin{bmatrix} 8 \\ 4 \end{bmatrix} = \begin{bmatrix} 17 \\ 8 \end{bmatrix},
$$

$$
(\mathbb{S} \otimes \nVdash_{\text{shift}})_{\mathrm{GEN},\star} = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 & 4 \\ 3 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 5 \end{bmatrix} + \begin{bmatrix} 8 \\ 5 \end{bmatrix} = \begin{bmatrix} 15 \\ 10 \end{bmatrix}.
$$

Interpretation: over the two days, ICU needs 17 *RNs and* 8 *LPNs in total; the General ward needs* 15 *RNs and* 10 *LPNs. The MetaMatrix organizes per–day, per–shift staffing as blocks while allowing standard block algebra for aggregation and downstream costing.*

Example 18 (MetaMatrix — university timetable (periods $\times$ rooms within departments $\times$ days))**.** *Let* $K = \{0, 1\}$ *(room occupancy). Rows index* departments $I = \{\mathrm{Math}, \mathrm{CS}\}$ *and columns index* days $J = \{\mathrm{Mon}, \mathrm{Tue}\}$*. Each block* $T_{ij} \in K^{4\times 3}$ *encodes a day's timetable for a department: rows are periods* $P = \{1, 2, 3, 4\}$ *and columns are rooms* $R = \{R_1, R_2, R_3\}$ *(1 if the room is used in that period).*

Define the MetaMatrix $\mathbb{T} : I \times J \to K^{4\times 3}$ *by*

$$
\mathbb{T} = \begin{array}{c|cc} & \text{Mon} & \text{Tue} \\ \hline \text{Math} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ \text{CS} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array} \in \prod_{(i,j)\in I\times J} K^{4\times 3}.
$$

Its flattening is a 8×6 *block–assembled occupancy matrix* $[\![\mathbb{T}]\!]$.

From occupancy to seated capacity. *Let room capacities be* $c(R_1) = 40$, $c(R_2) = 30$, $c(R_3) = 50$ *and form the block column with profile* $(J, \{\star\}; \mathbf{s}, \mathbf{t})$, *where* $\mathbf{s} = (3, 3)$, $\mathbf{t} = (1)$, *and*

$$
{}_{j\star} := \begin{bmatrix} 40 \\ 30 \\ 50 \end{bmatrix} \in \mathbb{R}^{3\times 1} \qquad (j = \mathrm{Mon}, \mathrm{Tue}).
$$

Then $(\mathbb{T} \otimes)_{i\star} = \sum_{j\in J} T_{ij}\, j_\star \in \mathbb{R}^{4\times 1}$ *returns the* seated capacity per period *accumulated over the two days for department* i. *For example, for* Math *on Monday alone,*

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 40 \\ 30 \\ 50 \end{bmatrix} = \begin{bmatrix} 90 \\ 30 \\ 40 \\ 50 \end{bmatrix},$$

so periods 1–4 *seat* $90, 30, 40, 50$ *students respectively. The MetaMatrix organizes each (department, day) timetable as a block, while block multiplication with per–day capacity vectors yields usable aggregates (per–period seat counts) without leaving matrix algebra.*

Definition 26 (Flattening (canonical block assembly))**.** *For a MetaMatrix* $\mathbb{A}$ *with profile* $(I, J; \mathbf{r}, \mathbf{s})$ *define its* flattening *(assembled block matrix)*

$$[\![\mathbb{A}]\!] \in K^{(\sum_i r_i)\times(\sum_j s_j)}$$

by placing each block A_{ij} *in its natural block position; i.e., rows are concatenated in the order* $i = 1, \ldots, p$ *and columns in the order* $j = 1, \ldots, q$.

Theorem 6 (Well-definedness and consistency with classical algebra)**.** *Let* $\mathbb{A}, \mathbb{B}$ *be MetaMatrices with the same profile* $(I, J; \mathbf{r}, \mathbf{s})$, *and let* $\mathbb{C}$ *have profile* $(J, L; \mathbf{s}, \mathbf{t})$ *compatible with* $\mathbb{A}$. *Then*

$$[\![\mathbb{A} \oplus \mathbb{B}]\!] = [\![\mathbb{A}]\!] + [\![\mathbb{B}]\!], \qquad [\![\lambda \odot \mathbb{A}]\!] = \lambda\, [\![\mathbb{A}]\!],$$

$$[\![\mathbb{A} \otimes \mathbb{C}]\!] = [\![\mathbb{A}]\!] \cdot [\![\mathbb{C}]\!].$$

In particular, $\oplus$, $\odot$, *and* $\otimes$ *are well defined and associative whenever the inner profiles match, and flattening is a homomorphism into the classical matrix algebra.*

Proof. All identities are standard block–matrix equalities. Each entry of $[\![\mathbb{A} \oplus \mathbb{B}]\!]$ (resp. $\lambda \odot \mathbb{A}$) equals the blockwise sum (resp. scalar multiple), which matches the definition of $\oplus$ (resp. $\odot$). For products, the (i, ℓ) block of the classical product equals $\sum_j A_{ij} B_{j\ell}$, which is exactly $(\mathbb{A} \otimes \mathbb{C})_{i\ell}$. Associativity and distributivity follow from the classical laws on blocks of compatible sizes. □

Theorem 7 (MetaMatrix as a MetaStructure; reduction to classical matrices)**.** *Let* Σ_{mat} *be the single–sorted signature with function symbols* $+$ *(binary),* $\cdot$ *(binary), and* $(\lambda\cdot)_{\lambda\in K}$ *(unary). For each pair* (r, s) *let*

$$\mathbf{M}_{r,s} := \big(K^{r\times s},\ +,\ \cdot\ (\textit{defined when } r = s),\ (\lambda\cdot)_{\lambda\in K}\big)$$

be the Σ_{mat}*–structure on the carrier* $K^{r\times s}$. *Fix a profile* $(I, J; \mathbf{r}, \mathbf{s})$ *and define the* MetaStructure operation

$$\Phi_{\oplus}\big(\mathbf{M}_{r_i,s_j}\big)_{(i,j)\in I\times J} := \Big(K^{(\sum_i r_i)\times(\sum_j s_j)},\ +,\ \cdot,\ (\lambda\cdot)\Big),$$

with $\Gamma_{\oplus}$ *assembling the carrier by block–concatenation and* $\Lambda_{\oplus}^{+}$ *prescribing blockwise addition; similarly define* $\Phi_{\cdot}$ *for block multiplication on compatible profiles via the classical block formula. Then:*

(a) $(U, \{\Phi_{\oplus}, \Phi_{\cdot}, (\Phi_\lambda)_{\lambda\in K}\})$ *with* $U = \{\mathbf{M}_{r,s} : r, s \in \mathbb{N}_{>0}\}$ *is a* MetaStructure *in the sense of Definition 14.*

(b) *The data of a MetaMatrix* $\mathbb{A}$ *with profile* $(I, J; \mathbf{r}, \mathbf{s})$ *is precisely the input tuple to* $\Phi_{\oplus}$, *and* $[\![\mathbb{A}]\!]$ *is the carrier produced by* $\Gamma_{\oplus}$. *The operations* $\oplus$, $\odot$, $\otimes$ *coincide with the meta-operations induced on that carrier.*

(c) *If* $r_i = s_j = 1$ *for all* i, j, *then every block is* 1×1 *and a MetaMatrix is exactly a classical matrix in* $K^{p\times q}$. *Thus MetaMatrix* generalizes *classical matrices.*

Proof. (a) Uniform carrier constructor Γ and symbol–constructors Λ are given by block assembly and the standard block formulas; naturality (isomorphism–invariance) is immediate from the functorial behavior of direct sums and products of vector spaces. (b) Unwinding the definitions shows that the meta-operations act blockwise exactly as in Definitions 25–26. (c) With 1×1 blocks, $[\![\cdot]\!]$ is the identity identification of entries with scalars, so we recover ordinary matrices and operations. □

Definition 27 (Depth, uniform profiles, and recursive objects). *Fix a* depth $t \in \mathbb{N}$ *and, for each level* $u = 1, \dots, t$, *fix a profile* $(I_u, J_u; \mathbf{r}^{(u)}, \mathbf{s}^{(u)})$. *A* depth–0 Iterated MetaMatrix *is a classical matrix* $A^{(0)} \in K^{r\times s}$ *(some* r, s*). Recursively, a* depth–u Iterated MetaMatrix $\mathbb{A}^{(u)}$ *is a MetaMatrix with profile* $(I_u, J_u; \mathbf{r}^{(u)}, \mathbf{s}^{(u)})$ *whose entries are depth–*$(u-1)$ *Iterated MetaMatrices, all using the same level–*$(u-1)$ *profile.*

Example 19 (Iterated MetaMatrix—regional advertising spend (Regions → Stores → Channels × DayTypes)). *We build a depth-2 Iterated MetaMatrix as in Definition 27.*

Level* 0 *(classical matrices). *Rows are* channels $C = \{\text{Online}, \text{InStore}\}$ *and columns are* day types $D = \{\text{Weekday}, \text{Weekend}\}$. *An entry records the (weekly) spend in USD. For a fixed store* S *and week* W, *a block is*

$$B^{(0)}_{\text{S,W}} = \begin{bmatrix} \textit{Online/Weekday} & \textit{Online/Weekend} \\ \textit{InStore/Weekday} & \textit{InStore/Weekend} \end{bmatrix} \in \mathbb{R}^{2\times2}_{\geq 0}.$$

Level* 1 *(MetaMatrix: Stores×Weeks). *Let* $I_1 = \{\text{S1}, \text{S2}\}$ *(stores),* $J_1 = \{\text{W1}, \text{W2}\}$ *(weeks), and* $\mathbf{r}^{(1)} = (2,2)$, $\mathbf{s}^{(1)} = (2,2)$ *(each block* 2×2*). For the (Region,Month)=* $(\text{East}, \text{Jan})$ *cell we specify the four level-0 blocks:*

$$\mathbb{A}^{(1)}_{\text{East,Jan}}(i,j) = B^{(0)}_{i,j} \quad \textit{with} \quad \begin{array}{l} B^{(0)}_{\text{S1,W1}} = \begin{bmatrix} 300 & 500 \\ 200 & 400 \end{bmatrix}, \; B^{(0)}_{\text{S1,W2}} = \begin{bmatrix} 320 & 480 \\ 220 & 380 \end{bmatrix}, \\ B^{(0)}_{\text{S2,W1}} = \begin{bmatrix} 250 & 450 \\ 180 & 350 \end{bmatrix}, \; B^{(0)}_{\text{S2,W2}} = \begin{bmatrix} 260 & 440 \\ 190 & 360 \end{bmatrix}. \end{array}$$

Level* 2 *(MetaMatrix: Regions×Months). *Let* $I_2 = \{\text{East}, \text{West}\}$ *and* $J_2 = \{\text{Jan}, \text{Feb}\}$ *with* $\mathbf{r}^{(2)} = (2,2)$, $\mathbf{s}^{(2)} = (2,2)$, *so each entry is a level-1 MetaMatrix as above. Thus* $\mathbb{A}^{(2)} : I_2 \times J_2 \to \text{MetaMat}_{(I_1, J_1; \mathbf{r}^{(1)}, \mathbf{s}^{(1)})}$.

Nested aggregation via MetaMatrix products. *Let* $d := \begin{bmatrix} 1 \\ 1 \end{bmatrix} \in \mathbb{R}^{2\times1}$ *(sum Weekday+Weekend). At level 1 define the block column* ${}^{(1)} : J_1 \times \{\star\} \to \mathbb{R}^{2\times1}$ *by* $({}^{(1)})_{j\star} = d$ *for each* $j \in J_1$. *Then the level-1 product*

$$\left(\mathbb{A}^{(1)}_{\text{East,Jan}} \otimes {}^{(1)}\right)_{i\star} = \sum_{j\in J_1} B^{(0)}_{i,j}\, d \in \mathbb{R}^{2\times1}$$

returns, for each store i, *the* two-channel weekly totals summed over weeks. *Compute explicitly:*

Store S1.

$$B^{(0)}_{\text{S1,W1}}\, d = \begin{bmatrix} 300+500 \\ 200+400 \end{bmatrix} = \begin{bmatrix} 800 \\ 600 \end{bmatrix}, \quad B^{(0)}_{\text{S1,W2}}\, d = \begin{bmatrix} 320+480 \\ 220+380 \end{bmatrix} = \begin{bmatrix} 800 \\ 600 \end{bmatrix},$$

$$\Rightarrow \sum_j B^{(0)}_{\text{S1},j}\, d = \begin{bmatrix} 1600 \\ 1200 \end{bmatrix}.$$

Store S2.

$$B^{(0)}_{\text{S2,W1}}\, d = \begin{bmatrix} 700 \\ 530 \end{bmatrix}, \quad B^{(0)}_{\text{S2,W2}}\, d = \begin{bmatrix} 700 \\ 550 \end{bmatrix}, \quad \Rightarrow \sum_j B^{(0)}_{\text{S2},j}\, d = \begin{bmatrix} 1400 \\ 1080 \end{bmatrix}.$$

Now sum channels by $\rho^\top := [1\ \ 1]$ *to obtain per-store monthly totals:*

$$\text{S1}: \ \rho^\top \begin{bmatrix} 1600 \\ 1200 \end{bmatrix} = 2800, \qquad \text{S2}: \ \rho^\top \begin{bmatrix} 1400 \\ 1080 \end{bmatrix} = 2480.$$

Finally, sum stores (scalar addition) to get the East–Jan regional total:

$$\textit{East–Jan monthly spend} \ = \ 2800 + 2480 \ = \ 5280 \ \textit{USD}.$$

(The same construction, applied one level higher as $\mathbb{A}^{(2)} \otimes {}^{(2)}$ *with* $({}^{(2)})_{j\star} = d$*, executes the week-summing step uniformly* inside *each level-1 block, illustrating the recursive nature.)*

Example 20 (Iterated MetaMatrix — manufacturing throughput (Regions → Plants → Lines×Days with Stations×Shifts blocks))**.** *We construct a depth-2 Iterated MetaMatrix capturing production counts.*

Level **0** ***(Stations×Shifts).*** *For each line/day, let rows be* shifts $S = \{\text{Day}, \text{Night}\}$ *and columns be* stations $R = \{\text{A}, \text{B}\}$*. An entry is the number of finished units. Thus a block* $P^{(0)}_{\text{Line,Day}} \in \mathbb{N}^{2\times 2}$.

Level **1** ***(MetaMatrix: Lines×Days).*** *Fix lines* $I_1 = \{\text{L1}, \text{L2}\}$ *and days* $J_1 = \{\text{Mon}, \text{Tue}\}$ *with* $\mathbf{r}^{(1)} = \mathbf{s}^{(1)} = (2, 2)$ *(each block* 2×2*). For* Plant North *we set:*

$$P^{(0)}_{\text{L1,Mon}} = \begin{bmatrix} 12 & 8 \\ 9 & 7 \end{bmatrix}, \ P^{(0)}_{\text{L1,Tue}} = \begin{bmatrix} 10 & 9 \\ 8 & 8 \end{bmatrix},$$
$$P^{(0)}_{\text{L2,Mon}} = \begin{bmatrix} 11 & 7 \\ 8 & 6 \end{bmatrix}, \ P^{(0)}_{\text{L2,Tue}} = \begin{bmatrix} 9 & 8 \\ 7 & 7 \end{bmatrix}.$$

For Plant South *we set:*

$$P^{(0)}_{\text{L1,Mon}} = \begin{bmatrix} 13 & 7 \\ 9 & 8 \end{bmatrix}, \ P^{(0)}_{\text{L1,Tue}} = \begin{bmatrix} 12 & 8 \\ 9 & 7 \end{bmatrix},$$
$$P^{(0)}_{\text{L2,Mon}} = \begin{bmatrix} 10 & 9 \\ 7 & 6 \end{bmatrix}, \ P^{(0)}_{\text{L2,Tue}} = \begin{bmatrix} 11 & 7 \\ 8 & 7 \end{bmatrix}.$$

Each plant thereby determines a level-1 MetaMatrix $\mathbb{P}^{(1)}_{\text{Plant}}$ *on* (I_1, J_1)*.*

Level **2** ***(MetaMatrix: Regions×Weeks).*** *Let regions* $I_2 = \{\text{North}, \text{South}\}$*, weeks* $J_2 = \{\text{W1}\}$*, and profile* $\mathbf{r}^{(2)} = \mathbf{s}^{(2)} = (2)$ *so that each* (Region, W1) *entry is the corresponding level-1* $\mathbb{P}^{(1)}_{\text{Plant}}$.

Nested aggregation (units per week). *Let* $e := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ *to sum stations, and* $\rho^\top := [1\ \ 1]$ *to sum shifts. At level 1 define* $({}^{(1)})_{j\star} = e$ *for* $j \in J_1$*. Then, for any line* i*,*

$$\left(\mathbb{P}^{(1)}_{\text{Plant}} \otimes {}^{(1)}\right)_{i\star} = \sum_{j \in J_1} P^{(0)}_{i,j}\, e \ \in \ \mathbb{N}^{2\times 1} \quad \textit{(per-shift totals, summed over days).}$$

Apply $\rho^\top$ *to obtain the line's two-day total. Compute explicitly:*

Plant North.

$$\text{L1}: \ P^{(0)}_{\text{L1,Mon}} e = \begin{bmatrix} 20 \\ 16 \end{bmatrix}, \ P^{(0)}_{\text{L1,Tue}} e = \begin{bmatrix} 19 \\ 16 \end{bmatrix} \ \Rightarrow \ \rho^\top(\cdot) = 36 + 35 = 71,$$
$$\text{L2}: \ P^{(0)}_{\text{L2,Mon}} e = \begin{bmatrix} 18 \\ 14 \end{bmatrix}, \ P^{(0)}_{\text{L2,Tue}} e = \begin{bmatrix} 17 \\ 14 \end{bmatrix} \ \Rightarrow \ \rho^\top(\cdot) = 32 + 31 = 63.$$

Plant North weekly total: $71 + 63 = 134$ *units.*

 doi:10.20944/preprints202510.0232.v1

Plant South.

$$\begin{aligned} \text{L1}: \; P^{(0)}_{\text{L1,Mon}} e &= \begin{bmatrix} 20 \\ 17 \end{bmatrix}, \; P^{(0)}_{\text{L1,Tue}} e = \begin{bmatrix} 20 \\ 16 \end{bmatrix} \;\Rightarrow\; \rho^\top(\cdot) = 37 + 36 = 73, \\ \text{L2}: \; P^{(0)}_{\text{L2,Mon}} e &= \begin{bmatrix} 19 \\ 13 \end{bmatrix}, \; P^{(0)}_{\text{L2,Tue}} e = \begin{bmatrix} 18 \\ 15 \end{bmatrix} \;\Rightarrow\; \rho^\top(\cdot) = 32 + 33 = 65. \end{aligned}$$

Plant South weekly total: $73 + 65 = 138$ *units.*

Regional aggregation (level 2). *Placing* $\mathbb{P}^{(1)}_{\text{North}}$ *and* $\mathbb{P}^{(1)}_{\text{South}}$ *as the two blocks of* $\mathbb{P}^{(2)}$ *(rows* I_2*, single column* J_2*) exposes a final summation across plants as a level-2 operation (scalar addition of the computed plant totals), yielding the region vector*

$$\begin{bmatrix} \textit{North} \\ \textit{South} \end{bmatrix} = \begin{bmatrix} 134 \\ 138 \end{bmatrix}.$$

This example illustrates how the same *block recipe ("sum columns by e, then sum rows by* $\rho^\top$*, then sum over days") is reused recursively inside each entry of the next level.*

Definition 28 (Recursive flattening and operations). *Define the* flattening $[\![\cdot]\!]^{\downarrow u}$ *by*

$$[\![A^{(0)}]\!]^{\downarrow 0} := A^{(0)}, \qquad [\![\mathbb{A}^{(u)}]\!]^{\downarrow u} := [\![\, \big([\![A^{(u-1)}_{ij}]\!]^{\downarrow(u-1)} \big)_{(i,j)\in I_u \times J_u}]\!],$$

i.e., first flatten each entry to a classical block, then assemble the block matrix. Define $\oplus$*,* $\odot$*, and* $\otimes$ *on depth–u objects entrywise at level u using the MetaMatrix rules, assuming inner profiles match.*

Theorem 8 (Flattening is a homomorphism at every depth). *For each depth* $u \geq 0$ *and all well–typed* $\mathbb{A}^{(u)}, \mathbb{B}^{(u)}$ *and scalars* λ*,*

$$[\![\mathbb{A}^{(u)} \oplus \mathbb{B}^{(u)}]\!]^{\downarrow u} = [\![\mathbb{A}^{(u)}]\!]^{\downarrow u} + [\![\mathbb{B}^{(u)}]\!]^{\downarrow u}, \qquad [\![\lambda \odot \mathbb{A}^{(u)}]\!]^{\downarrow u} = \lambda \, [\![\mathbb{A}^{(u)}]\!]^{\downarrow u},$$

$$[\![\mathbb{A}^{(u)} \otimes \mathbb{B}^{(u)}]\!]^{\downarrow u} = [\![\mathbb{A}^{(u)}]\!]^{\downarrow u} \cdot [\![\mathbb{B}^{(u)}]\!]^{\downarrow u}.$$

Proof. Induction on u. The base $u = 0$ is trivial. For $u \to u+1$, apply the induction hypothesis to each entry (depth u), then Theorem 6 at the top MetaMatrix level to assemble blocks; the three identities follow. □

Theorem 9 (Iterated MetaMatrix as an Iterated MetaStructure; reductions). *For each level u let* $U^{(u)}$ *be the class of depth–u Iterated MetaMatrices with fixed profile* $(I_u, J_u; \mathbf{r}^{(u)}, \mathbf{s}^{(u)})$*. The triple*

$$\mathfrak{M}^{(u)} := \big(U^{(u)}, \, \Phi^{(u)}_{\oplus}, \, \Phi^{(u)}_{\cdot}, \, (\Phi^{(u)}_{\lambda})_{\lambda \in K} \big),$$

where $\Phi^{(u)}$ *applies the MetaMatrix constructors to entries in* $U^{(u-1)}$ *and then assembles via* Γ *(Definition 14), is an* Iterated MetaStructure*. Moreover:*

(a) (Generalization) *Depth* 1 *recovers MetaMatrix; depth* 0 *recovers classical matrices.*
(b) (Compatibility) *The flattening* $[\![\cdot]\!]^{\downarrow u}$ *is a natural homomorphism* $\mathfrak{M}^{(u)} \to K\text{–Mat}$ *(Theorem 8).*

Proof. The carrier constructors Γ and symbol–constructors Λ are given uniformly at each depth by the block assembly of Definition 28 and the block rules of Definition 25. Naturality is inherited from the functoriality of assembling direct sums/products of vector spaces. (a) follows from the definitions; (b) is Theorem 8. □

5. Conclusions

In this paper we defined *HyperMatrix*, *SuperHyperMatrix*, *MultiMatrix*, *Iterative MultiMatrix*, *MetaMatrix*, and *Iterated MetaMatrix*—all as extensions of the classical notion of a matrix—and we offer a concise examination of their properties. In future work, we plan to consider extensions that incorporate uncertainty and multi-valuedness by employing advanced set-theoretic frameworks such as the *Fuzzy Set* [49–52], *Intuitionistic Fuzzy Set* [53,54], Vague Sets [55–57], *Hesitant Fuzzy Set* [58–60], Picture Fuzzy Set [61–63], *Neutrosophic Set* [64–66], and *Plithogenic Set* [67–70].

Funding: No external funding was received for this work.

Institutional Review Board Statement: This research did not involve human participants or animals, and therefore did not require ethical approval.

Data Availability Statement: 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.

Use of Artificial Intelligence: 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.

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.

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

References

1. Smarandache, F. Foundation of SuperHyperStructure & Neutrosophic SuperHyperStructure. *Neutrosophic Sets and Systems* **2024**, *63*, 21.
2. Al-Tahan, M.; Davvaz, B.; Smarandache, F.; Anis, O. On some neutroHyperstructures. *Symmetry* **2021**, *13*, 535.
3. Davvaz, B.; Nezhad, A.D.; Nejad, S.M. Algebraic hyperstructure of observable elementary particles including the Higgs boson. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences* **2020**, *90*, 169–176.
4. Ruggero, M.S.; Vougiouklis, T. Hyperstructures in Lie-Santilli admissibility and iso-theories. *Ratio Mathematica* **2017**, *33*, 151.
5. Das, A.K.; Das, R.; Das, S.; Debnath, B.K.; Granados, C.; Shil, B.; Das, R. A Comprehensive Study of Neutrosophic SuperHyper BCI-Semigroups and their Algebraic Significance. *Transactions on Fuzzy Sets and Systems* **2025**, *8*, 80.
6. Rahmati, M.; Hamidi, M. Extension of g-algebras to superhyper g-algebras. *Neutrosophic Sets and Systems* **2023**, *55*, 557–567.
7. Fujita, T. Weak HyperFuzzy Set and Weak SuperHyperFuzzy Set. *Computational Methods* **2025**, *2*, 11–26.
8. Kong, X. Students' Innovation Ability Evaluation in Vocational Colleges Based on the STEAM Education Concept: A Neutrosophic SuperHyper Pentapartitioned Process Model. *Neutrosophic Sets and Systems* **2025**, *93*, 250–261.
9. Fujita, T. Medical SuperHyperStructure and Healthcare SuperHyperStructure. *Journal of Medicine and Health Research* **2025**, *10*, 243–262.
10. Li, Y. A Neutrosophic SuperHyperModal Framework for the Multilayered Evaluation of New Media Marketing Strategies in the Automotive Industry. *Neutrosophic Sets and Systems* **2025**, *88*, 28.
11. Jech, T. *Set theory: The third millennium edition, revised and expanded*; Springer, 2003.
12. Smarandache, F. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*; Infinite Study, 2023.
13. Smarandache, F. Introduction to SuperHyperAlgebra and Neutrosophic SuperHyperAlgebra. *Journal of Algebraic Hyperstructures and Logical Algebras* **2022**.
14. Fujita, T.; Smarandache, F. A Unified Framework for U-Structures and Functorial Structure: Managing Super, Hyper, SuperHyper, Tree, and Forest Uncertain Over/Under/Off Models. *Neutrosophic Sets and Systems* **2025**, *91*, 337–380.
15. Selvachandran, G.; Salleh, A.R. Hypergroup theory applied to fuzzy soft sets. *Global Journal of Pure and Applied Sciences* **2015**, *11*, 825–834.

doi:10.20944/preprints202510.0232.v1

16. Stenstrom, B. *Rings of quotients: an introduction to methods of ring theory*; Vol. 217, Springer Science & Business Media, 2012.
17. Jun, Y.B.; Park, C.H.; Alshehri, N.O. Hypervector spaces based on intersectional soft sets. In Proceedings of the Abstract and Applied Analysis. Wiley Online Library, 2014, Vol. 2014, p. 784523.
18. Tallini, M.S. Hypervector spaces. In Proceedings of the Proceeding of the 4th International Congress in Algebraic Hyperstructures and Applications, 1991, pp. 167–174.
19. Gross, J.L.; Yellen, J.; Anderson, M. *Graph theory and its applications*; Chapman and Hall/CRC, 2018.
20. Diestel, R. Graph theory 3rd ed. *Graduate texts in mathematics* **2005**, *173*, 12.
21. Diestel, R. *Graph theory*; Springer (print edition); Reinhard Diestel (eBooks), 2024.
22. Vougioukli, S. Helix hyperoperation in teaching research. *Science & Philosophy* **2020**, *8*, 157–163.
23. Vougioukli, S. Hyperoperations defined on sets of S -helix matrices. 2020.
24. Rezaei, A.; Smarandache, F.; Mirvakili, S. Applications of (Neutro/Anti)sophications to Semihypergroups. *Journal of Mathematics* **2021**.
25. Vougioukli, S. HELIX-HYPEROPERATIONS ON LIE-SANTILLI ADMISSIBILITY. *Algebras Groups and Geometries* **2023**.
26. Corsini, P.; Leoreanu, V. *Applications of hyperstructure theory*; Vol. 5, Springer Science & Business Media, 2013.
27. Novák, M.; Křehlík, Š.; Ovaliadis, K. Elements of hyperstructure theory in UWSN design and data aggregation. *Symmetry* **2019**, *11*, 734.
28. Oguz, G.; Davvaz, B. Soft topological hyperstructure. *J. Intell. Fuzzy Syst.* **2021**, *40*, 8755–8764.
29. Smarandache, F. SuperHyperStructure & Neutrosophic SuperHyperStructure, 2024. Accessed: 2024-12-01.
30. Shinoj, T.; John, S.J. Intuitionistic fuzzy multisets and its application in medical diagnosis. *World academy of science, engineering and technology* **2012**, *6*, 1418–1421.
31. Ejegwa, P. New operations on intuitionistic fuzzy multisets. *Journal of Mathematics and Informatics* **2015**, *3*, 17–23.
32. Girish, K.; John, S.J. Rough Multisets and Information Multisystems. *Advances in Decision sciences* **2011**.
33. Girish, K.; John, S.J. On rough multiset relations. *International Journal of Granular Computing, Rough Sets and Intelligent Systems* **2014**, *3*, 306–326.
34. Fujita, T. Iterative Multistructure and Curried Iterative Multistructure: Graph, Function, Chemical Structure, and beyond **2025**.
35. Fujita, T. Introduction for Structure, HyperStructure, SuperHyperStructure, MultiStructure, Iterative MultiStructure, TreeStructure, and ForestStructure **2025**.
36. Fujita, T. MetaStructure, Meta-HyperStructure, and Meta-SuperHyperStructure, 2025. Preprint, https://doi.org/10.13140/RG.2.2.22609.03680.
37. Fujita, T. MetaHyperGraphs, MetaSuperHyperGraphs, and Iterated MetaGraphs: Modeling Graphs of Graphs, Hypergraphs of Hypergraphs, Superhypergraphs of Superhypergraphs, and Beyond, 2025. https://doi.org/10.31224/5103.
38. Horn, R.A.; Johnson, C.R. Matrix analysis. 1985.
39. Yaz, E.E. Linear Matrix Inequalities In System And Control Theory. *Proceedings of the IEEE* **1998**, *86*, 2473–2474.
40. Golub, G.H. Matrix computations. 1983.
41. Chrzeszczyk, A.; Kochanowski, J. Matrix Computations. In Proceedings of the Encyclopedia of Parallel Computing, 2011.
42. Wu, F.; Zhang, T.; de Souza, A.H.; Fifty, C.; Yu, T.; Weinberger, K.Q. Simplifying Graph Convolutional Networks. In Proceedings of the International Conference on Machine Learning, 2019.
43. Lee, D.D.; Sebastian, H.; y, S. Algorithms for Non-negative Matrix Factorization. In Proceedings of the Neural Information Processing Systems, 2000.
44. Engler, A.J.; Engler, A.J.; Sen, S.; Sen, S.; Sweeney, H.L.; Discher, D.E. Matrix Elasticity Directs Stem Cell Lineage Specification. *Cell* **2006**, *126*, 677–689.
45. Motameni, M.; Ameri, R.; Sadeghi, R. Hypermatrix Based on Krasner Hypervector Spaces. 2013.
46. Fujita, T. A theoretical exploration of hyperconcepts: Hyperfunctions, hyperrandomness, hyperdecision-making, and beyond (including a survey of hyperstructures). *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond* **2025**, p. 111.
47. Tropp, J.A. An Introduction to Matrix Concentration Inequalities. *ArXiv* **2015**, *abs/1501.01571*.
48. Bellman, R. Introduction to Matrix Analysis. 1972.
49. Zadeh, L.A. Fuzzy sets. *Information and control* **1965**, *8*, 338–353.

50. Akram, M.; Luqman, A. *Fuzzy hypergraphs and related extensions*; Springer, 2020.
51. Al-Hawary, T. Complete fuzzy graphs. *International Journal of Mathematical Combinatorics* **2011**, *4*, 26.
52. Nishad, T.; Al-Hawary, T.A.; Harif, B.M. General Fuzzy Graphs. *Ratio Mathematica* **2023**, *47*.
53. Atanassov, K.T. Circular intuitionistic fuzzy sets. *Journal of Intelligent & Fuzzy Systems* **2020**, *39*, 5981–5986.
54. Atanassov, K.T.; Gargov, G. *Intuitionistic fuzzy logics*; Springer, 2017.
55. Lu, A.; Ng, W. Vague sets or intuitionistic fuzzy sets for handling vague data: which one is better? In Proceedings of the International conference on conceptual modeling. Springer, 2005, pp. 401–416.
56. Gau, W.L.; Buehrer, D.J. Vague sets. *IEEE transactions on systems, man, and cybernetics* **1993**, *23*, 610–614.
57. Bustince, H.; Burillo, P. Vague sets are intuitionistic fuzzy sets. *Fuzzy sets and systems* **1996**, *79*, 403–405.
58. Torra, V.; Narukawa, Y. On hesitant fuzzy sets and decision. In Proceedings of the 2009 IEEE international conference on fuzzy systems. IEEE, 2009, pp. 1378–1382.
59. Torra, V. Hesitant fuzzy sets. *International journal of intelligent systems* **2010**, *25*, 529–539.
60. Xu, Z. *Hesitant fuzzy sets theory*; Vol. 314, Springer, 2014.
61. Das, S.; Ghorai, G.; Pal, M. Picture fuzzy tolerance graphs with application. *Complex & Intelligent Systems* **2022**, *8*, 541–554.
62. Khan, W.A.; Arif, W.; NGUYEN, Q.H.; Le, T.T.; Van Pham, H. Picture Fuzzy Directed Hypergraphs with applications towards decision-making and managing hazardous chemicals. *IEEE Access* **2024**.
63. Cuong, B.C.; Kreinovich, V. Picture fuzzy sets-a new concept for computational intelligence problems. In Proceedings of the 2013 third world congress on information and communication technologies (WICT 2013). IEEE, 2013, pp. 1–6.
64. Smarandache, F.; Jdid, M. An Overview of Neutrosophic and Plithogenic Theories and Applications **2023**.
65. Broumi, S.; Talea, M.; Bakali, A.; Smarandache, F. Single valued neutrosophic graphs. *Journal of New theory* **2016**, *10*, 86–101.
66. Broumi, S.; Talea, M.; Bakali, A.; Smarandache, F.; Kumar, P.K. Shortest path problem on single valued neutrosophic graphs. In Proceedings of the 2017 international symposium on networks, computers and communications (ISNCC). IEEE, 2017, pp. 1–6.
67. Sultana, F.; Gulistan, M.; Ali, M.; Yaqoob, N.; Khan, M.; Rashid, T.; Ahmed, T. A study of plithogenic graphs: applications in spreading coronavirus disease (COVID-19) globally. *Journal of ambient intelligence and humanized computing* **2023**, *14*, 13139–13159.
68. Kandasamy, W.V.; Ilanthenral, K.; Smarandache, F. *Plithogenic Graphs*; Infinite Study, 2020.
69. Sathya, P.; Martin, N.; Smarandache, F. Plithogenic Forest Hypersoft Sets in Plithogenic Contradiction Based Multi-Criteria Decision Making. *Neutrosophic Sets and Systems* **2024**, *73*, 668–693.
70. Smarandache, F. *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.