symmetry

MDPI

Article

Elements of Hyperstructure Theory in UWSN Design and Data Aggregation

Michal Novák [1,*], Štepán Křehlík [2] and Kyriakos Ovaliadis [3]

1 Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 8, 616 00 Brno, Czech Republic
2 Department of Applied Mathematics and Computer Science, Masaryk University, Lipová 41a, 602 00 Brno, Czech Republic; Stepan.Krehlik@econ.muni.cz
3 Department of Electrical Engineering, Eastern Macedonia and Thrace Institute of Technology, Agios Loukas, 654 04 Kavala, Greece; ovaliad@teiemt.gr
* Correspondence: novakm@feec.vutbr.cz; Tel.: +420-54114-6077

Received: 8 May 2019; Accepted: 27 May 2019; Published: 29 May 2019

check for updates

Abstract: In our paper we discuss how elements of algebraic hyperstructure theory can be used in the context of underwater wireless sensor networks (UWSN). We present a mathematical model which makes use of the fact that when deploying nodes or operating the network we, from the mathematical point of view, regard an operation (or a hyperoperation) and a binary relation. In this part of the paper we relate our context to already existing topics of the algebraic hyperstructure theory such as quasi-order hypergroups, EL-hyperstructures, or ordered hyperstructures. Furthermore, we make use of the theory of quasi-automata (or rather, semiautomata) to relate the process of UWSN data aggregation to the existing algebraic theory of quasi-automata and their hyperstructure generalization. We show that the process of data aggregation can be seen as an automaton, or rather its hyperstructure generalization, with states representing stages of the data aggregation process of cluster protocols and describing available/used memory capacity of the network.

Keywords: clustering protocols; quasi-automaton; quasi-multiautomaton; semihypergroup; UWSN

1. Introduction

Underwater wireless sensor networks (UWSN) are often used in environment monitoring where they review how human activities affect marine ecosystems, undersea explorations such as detecting oilfields, for disaster prevention, e.g., when monitoring ocean currents, in assisted navigation for the location of dangerous rocks in shallow waters, or for disturbed tactical surveillance for intrusion detection.

The fact that such wireless sensor networks are deployed underwater results in profound differences from terrestrial wireless sensor networks. The key aspects that are different include the communication method, i.e., radio waves vs acoustic signals, cost (while terrestrial networks experience decreasing prices of components, underwater sensors are still expensive devices), memory capacity (because water is a problematic medium resulting in the loss of large quantities of data), power limitations due to the nature of the signal and longer distances handled, as well as problems related to the deployment of the network, i.e., issues connected to static or dynamic deployment. In underwater sensor networks, we commonly face challenges of limited bandwith, high bit error rates, large propagation delays, and limited battery resources caused by the fact that in an underwater environment, sensor batteries are impossible to recharge especially because no solar energy is available underwater. The power losses, which cannot be avoided, result in the need to reconfigure the network topology in order to maintain network connectivity and communication between sensor nodes. Thus, size of the UWSN coverage area and efficiency of data aggregation are affected. Obviously, efficiency in battery use influences network lifetime without sacrificing system performances. These differences are shown in Table 1.

Table 1. Comparison of some features of terrestrial and underwater wireless sensor networks (UWSN).

	(Terrestrial) WSN	UWSN
Communication Media	RF Waves	Acoustic Waves
Frequency	High	Low
Node size	Small	Large
Deployment	Dense	Sparse
Power	Low	High
Energy consumption	Low	High
Propagation delay	Low	High
Bandwidth	High	Low
Path loss	Low	High
Cost	Inexpensive	Expensive
Memory	Sensor nodes have low capacity	Sensor nodes require large capacity

We use different protocols for discovering and maintaining routes between sensor nodes. As mentioned in Novák, Křehlík, and Ovaliadis [1], the most commonly used routing protocols are: Flooding, multipath, cluster, and miscellaneous protocols, see Wahid and Dongkyun [2]. In the flooding approach, the transmitters send a packet to all nodes within the transmission range. In the multipath approach, source sensor nodes establish more than one path towards sink nodes on the surface. Finally, in the clustering approach the sensor nodes are grouped together in a cluster. For an easy-to-follow reading on how UWSN's work and on advantages of clustering see Domingo and Prior [3], the basic idea is shown in Figures 1, 2.

Recent research shows that the cluster based protocols give a great contribution towards the concept of energy efficient networks, see Ayaz et al. [4], Ovaliadis and Savage [5], or Rault, Abdelmadjid, and Yacine [6]. A common cluster based network consists of a centralized station deployed at the surface of the sea called a sink (or surface station) and sensor nodes deployed at various tiers inside the sea environment. These are grouped into clusters. In this architecture, each cluster has a head sensor node called a cluster head (CH). The cluster head is assumed to be inside the transmission range of all sensor nodes that belong to its cluster. Every cluster head operates as a coordinator for its cluster, performing significant tasks such as cluster maintenance, transmission arrangements, data aggregation, and data routing (Figure 2).

Mathematical Background of the Model

In the UWSN topology, several aspects are important for successful data aggregation. First of all, there must exist a path linking every element of the network to the surface station. However, these paths need not be unique as there might be multiple possible paths which the data from a given element can use to reach the surface station. Second, there always exists a cetain kind of ordering of the set of the network elements. They can be ordered with respect to their physical depth, with respect to their importance, with respect to communication priority, remaining battery power, etc. Finally, as data are collected, they are combined in the "upwards" elements in order to be sent further on.

Thus one may employ techniques of algebra or graph theory in the description of the data aggregation process as has been recently done by Aboyamita et al., Domingo, or Jiang et al. [7–9]. However, given the multivalued nature of data aggregation (multiple paths, more than one possible links of elements, etc.), it seems relevant to make use of the elements of the algebraic hyperstructure theory. Notice that while in "classical" algebra, we regard operations, i.e., mappings $f : H^n \to H$, in the algebraic hyperstructure theory we work with hyperoperations, i.e., mappings $g : H^n \to \mathcal{P}^*(H)$, where $\mathcal{P}^*(H)$ is the power set of H with $\emptyset$ excluded (one need not consider this exclusion though). For the general introduction to the theory as well as definitions of concepts not explicitly defined further on, see Corsini and Leoreanu [10].

In the algebraic hyperstructure theory, there are several concepts which make use of the aspect of ordering. A small selection includes Comer, Corsini, Cristea, De Salvo et al. [11–15]. Further on we discuss three of these: EL–hyperstructures, quasi-order hypergroups, and ordered hyperstructures. Each of these concepts uses somewhat different background and assumptions:

EL–hyperstructures are constructed from pre- and partially-ordered semigroups, i.e., the hyperoperation is defined using an operation and a relation compatible with it;
Quasi-order hypergroups are constructed from pre-ordered sets, i.e., the hyperoperation is defined using a relation only;
Ordered hyperstructures are algebraic hyperstructures on which a relation compatible with the hyperoperation is defined.

All of these have been studied in depth and numerous results have been achieved in their respective theories. The idea of EL–hyperstructures has been implicitely present in a number of works since at least the 1960s, for example Pickett [16]. The definition and first results were given by Chvalina [17] and the theory has been elaborated by Novák (later jointly with Chvalina, Křehlík, and Cristea) in a series of papers including [18–22]. It is to be noted that, since the class of EL–hyperstructures is rather broad, the aim of many theorems included in some of those papers was to establish a common ground for some already existing ad hoc derived results. Recently, some examples concerning various types of cyclicity in hypergroups have been constructed using EL–hyperstructures, see Novák, Křehlík and Cristea [23].

The idea of quasi-order hypergroups was proposed by Chvalina in [17,24,25]. Some results achieved with the help of this concept are included in Corsini and Leoreanu [10]. Not to be missed are results concerning the theory of automata collected in Chvalina and Chvalinová [25]. It should be stressed that these results were motivated by Comer [26] and Massouros and Mittas [27].

Ordered hyperstructures were introduced by Heidari and Davvaz [28]. Numerous results have been published since, mainly by Iranian authors.

For the following set of basic definitions see Novák, Křehlík, and Ovaliadis [1].

Definition 1. *By an EL–semihypergroup we mean a semihypergroup, in which, for all $a, b \in H$, there is $a * b = \{x \in H \mid a \cdot b \leq x\}$, where $(H, \cdot, \leq)$ is a quasi-ordered semigroup.*

Proposition 1. *[20,22] If, for all $a, b \in H$, there is $\{a, b\} \in a * b$, then the EL–semihypergroup $(H, *)$ is a hypergroup. If $(H, \cdot, \leq)$ is a partially ordered group, then its EL–hypergroup $(H, *)$ is a join space.*

Definition 2. *Let $(H, *)$ be a hypergroupoid. We say that H is a quasi-order hypergroup, i.e., a hypergroup determined by a quasi-order, if, for all $a, b \in H$, $a \in a^3 = a^2$, and $a * b = a^2 \cup b^2$. Moreover, if $a^2 = b^2 \Rightarrow a = b$ holds for all $a, b \in H$, then $(H, *)$ is called an order hypergroup.*

Proposition 2. *[10] A hypergroupoid is a quasi-order hypergroup if and only if there exists a quasi-order "$\leq$" on the set H such that, for all $a, b \in H$, there is $a * b = [a)_\leq \cup [b)_\leq$.*

Definition 3. *An ordered semihypergroup $(H, *, \preceq)$ is a semihypergroup $(H, *)$ together with a partial ordering "$\preceq$" which is compatible with the hyperoperation, i.e., $x \preceq y \Rightarrow a * x \preceq a * y$ and $x * a \preceq y * a$ for all $a, x, y \in H$. By $a * x \preceq a * y$ we mean that for every $c \in a * x$ there exists $d \in a * y$ such that $c \preceq d$.*

Notation. Further on, for some $a \in H$, by $[a)_\leq$ means the set $\{x \in H \mid a \leq x\}$. For this reason, closed intervals will not be denoted by $[a, b]$ but by $\langle a; b\rangle$.

2. Mathematical Model

The mathematical model presented in this section was published as an extended abstract of the conference contribution Novák, Ovaliadis, and Křehlík [1] presented by the authors of this paper at International Conference on Numerical Analysis and Applied Mathematics (ICNAAM 2017).

UWSNs consist of elements of different types: First, we have *surface stations*, which pass data to a ship or to a data-collecting station located on the sea shore; second, we have *sensor nodes* deployed at various tiers in water or at the sea bed. The sensors, which are deployed in water, can function as sensors measuring the requested data or as transporters of information from seabed sensors. In any case, information collected from all sensors must be passed to surface stations. From these it can be collected either by a ship passing by or, alternatively, transmitted to a data-collecting station located on the sea shore. The ship or the data-collecting stations are *central nodes*.

Denote H the set of all elements of an arbitrary UWSN. Suppose that all elements are capable of handling (i.e., receiving or transmitting) data in the same way. Also suppose that they perform the same set of tasks. Thus they are, from the mathematical point of view, interchangeable and equal (of course, with respect to their functionality as sinks and sensor nodes). The aim of the system is to collect information. Therefore, our elements of H must communicate data. This should be done ideally upwards, towards the surface. As we have mentioned above, there are different ways of passing information. In our model we concentrate on multipath and cluster routing approach (see Figure 1 and Figure 2). For details concerning these see Ayaz et al. and Li et al. [4,29]. Multipath routing protocols (Figure 1), forward the data packets to the sink via other nodes while in cluster based routing protocols (Figure 2), data packets are first aggregated to the respective cluster heads and only then forwarded via other cluster heads to the sink. For our purposes, we denote the i–th cluster by cl_i. Its cluster head will be denoted by CH_i. We call non-CH nodes *ordinary* and sinks will be treated as cluster heads.

Now, suppose that the elements of our system are clustered. In other words, some elements of H function as cluster heads, i.e., masters, while others are ordinary. The data aggregation process goes as follows: Within their cluster, the ordinary elements pass information to their cluster head while between clusters, i.e., supposedly over longer distances, only cluster heads communicate. At a given point in time, each cluster has the unique cluster head, and each element can belong to exactly one cluster. We denote the i–th cluster by cl_i and its cluster head by CH_i.

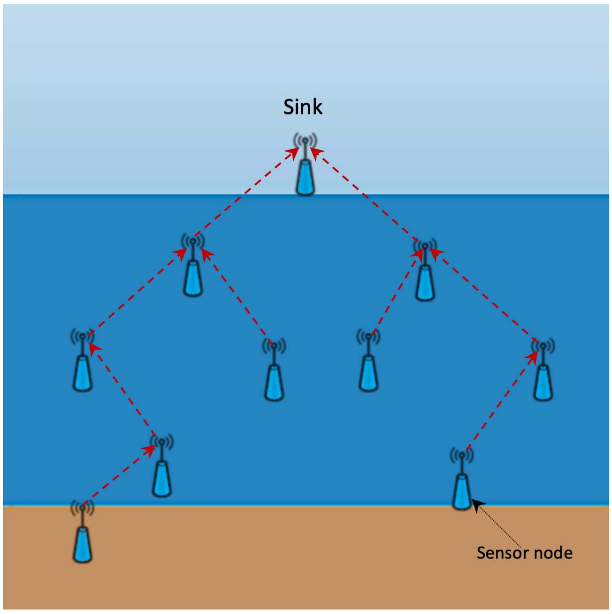

Figure 1. Multipath approach to UWSN data aggregation. Notice the oriented communication between nodes.

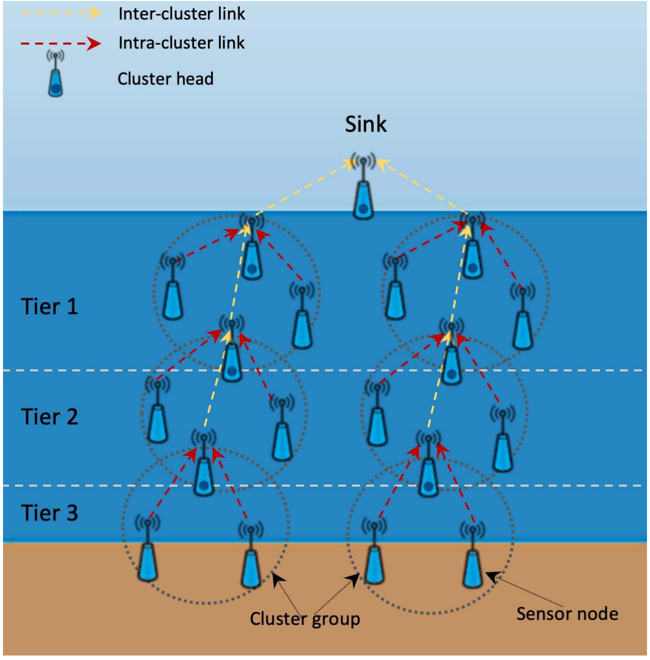

Figure 2. Cluster based approach to UWSN data aggregation—idealized deployment. The tiers need not be horizontal, we usually regard distance towards sink instead of depth.

Now, for a given pair $a, b \in H$, regard a binary hyperoperation, where $a * b$ is, for arbitrary $a, b \in H$, defined by:

$$a * b = \begin{cases} \{a,b\} \cup [a \cdot b)_{\leq} & \text{for } (a = CH_i, b = CH_j) \text{ or } a, b \in cl_i \\ \{a,b\} & \text{for } (a \neq CH_i \text{ or } b \neq CH_j) \text{ and } (a \in cl_i, b \in cl_j, i \neq j) \end{cases} \quad (1)$$

By $[a \cdot b)_{\leq}$ we mean a set $\{x \in H \mid a \cdot b \leq x\}$, where $a \cdot b$ is a result of a single-valued binary operation such that $a \cdot b$ is, for arbitrary $a, b \in H$, defined by:

$$a \cdot b = \begin{cases} CH_i & \text{for } a, b \in cl_i \\ CH_k & \text{for } a = CH_i, b = CH_j, i \neq j \\ s & \text{for } ((a \neq CH_i \text{ or } b \neq CH_j) \text{ and } (a \in cl_i, b \in cl_j, i \neq j)) \text{ or } a = s \text{ or } b = s \end{cases} \quad (2)$$

and CH_k is such a cluster head that $CH_i \leq CH_k$, $CH_j \leq CH_k$, where $a \leq b$ is a relation between elements of H such that: (1) $s \leq s$, $s \leq CH_i$ and $CH_i \leq s$ for all clusters cl_i, (2) within the same cluster cl_i we have $a_j \leq CH_i$ for all $a_j \in cl_i$ while mutually different ordinary elements of the cluster are incomparable, (3) between clusters for $a = CH_i$, $b = CH_j$ the fact that $a \leq b$ means that the tier of b (measured towards the surface) is smaller than or equal to the tier of a, and (4) in all other cases a and b are not related. By CH_k above we mean a cluster head on the closest tier above both CH_i and CH_j. Of course, CH_k always exists yet need not be unique as there may be more cluster heads at this closest tier. In such a case, we choose the most suitable one or regard all cluster heads as equal. Notice that, in our definitions, the fact that $CH_i \leq CH_j$ and simultaneously $CH_j \leq CH_i$ does not mean that $CH_i = CH_j$, rather it only means that CH_i and CH_j are on the same tier. If we are able to chose the most suitable cluster head (further on we remark that we are), the relation "$\leq$" (restricted to $H \setminus \{s\}$) becomes partial ordering and we can write $CH_k = \sup\{CH_i, CH_j\}$ (with respect to the relation "$\leq$"). Finally, the element s is an element of H reserved for situations when a and b fail to communicate. It is artificially added to our set of elements H or we can agree that one (given the actual sensor deployment is of course carefully chosen) of elements of H will be s. In this way, technically speaking, we should in fact write $H_e = H \cup \{s\}$, where H_e could mean "expanded". Of course, if we choose the option of $s \in H$, then $H_e = H$.

Under these definitions, $a \cdot b$ is the element in which the data from a and b meet, and $a * b$ is the path in which the data from both a and b can spread. The facts that $a \cdot b = s$ or $a * b = \{a, b\}$ or $a * b = \{a, b, s\}$ all stand for communication failure.

Lemma 1. [1] *$(H, \leq)$ is a quasi-ordered set.*

Suppose now that we have arbitrary $a, b \in H$. Since the result of $a \cdot b$ is such an element of H in which the data from a and b meet, it is natural to suppose that $a \cdot b = b \cdot a$, i.e., that "$\cdot$" is commutative. However, we can suppose this only on condition that there exists such an algorithms that $a \cdot b = CH_k = CH_l = b \cdot a$ for arbitrary clusters cl_k, cl_l. Further on suppose that such an algorithm exists, i.e., that $(H, \cdot)$ is a commutative groupoid. The following lemma is obvious.

Lemma 2. [1] *If $(H, \cdot)$ is a commutative groupoid, then $(H, *)$ is a commutative hypergroupoid.*

In the following lemma notice that weak associativity of the hyperoperation is defined as $a * (b * c) \cap (a * b) * c \neq \emptyset$ for all $a, b, c \in H$; a quasi-hypergroup is a reproductive hypergroupoid.

Lemma 3. [1] *The hypergroupoid $(H, *)$ is a H_v–group, i.e., a weak associative quasi-hypergroup.*

Lemma 4. [1] *The quasi-ordering "$\leq$" and the operation "$\cdot$" are compatible, i.e., for all $a, b \in H$ such that $a \leq b$ and an arbitrary $c \in H$ there is $a \cdot c \leq b \cdot c$ and $c \cdot a \leq c \cdot b$.*

Now, denote $H_{CH} \subseteq H$ the set of cluster heads. This notation enables us to regard both clustering based systems and multipath systems because the fact that $H_{CH} = H$ means that every element of H is a cluster head, i.e., the system is in fact a multipath one. In such a case the model simplifies substantially. This is because there is no need for the special element s and we do not distinguish between communication within and between clusters. The operation "$\cdot$" defined by Equation (2) reduces to $a \cdot b = c$ (we still suppose that it is commutative) and, consequently, the hyperoperation Equation (1) reduces to $a * b = \{a, b\} \cup [a \cdot b)_{\leq}$, in both cases for all $a, b \in H$.

Lemma 5. [1] *If we are able to uniquely identify CH_k in Equation (2), then $(H_{CH}, \cdot, \leq)$ is a partially ordered semigroup.*

Finally, what is $x \in [a)_{\leq}$? This means that $a \leq x$, i.e., that the data from the element a reach the element x. Thus, if x is a sink, than the fact that $x \in [a)_{\leq}$ means that the data from a can be successfully collected. What we want is that, if we denote S the set of all sinks, for all $a \in H$ there exists at least one $x \in S$ such that $x \in [a)_{\leq}$, which means that data from all elements of our network H can be successfully collected. Of course, in order to achieve this, it is crucial to have an algorithm for unique determination of CH_k in Equation (2). Yet clustering algorithms such as the Distributed Underwater Clustering Scheme (DUCS) [3] or Low Energy Adaptive Clustering Hierarchy (LEACH) protocol can provide this.

3. Use of the Theory of Quasi-Automata

In Definition 2, the concept of quasi-order hypergroup is defined. Chvalina and Chvalinová [25] relate these to the theory of quasi-automata, i.e., automata without output. For an automaton they construct a quasi-order hypergroup of its state set and show that the automaton is connected if and only if the state hypergroup is inner irreducible as well as strogly connected, i.e., we can reach any state from any other state, if and only if the state hypergroup is (in a special way) cyclic. In other words, if we look at the problem of data aggregation from the point of view of the automata theory, where every step is an application of the transition function with the initial state "data aggregation to begin" and the desirable state "data from all elements collected" (or rather "useful data from all elements sent" since every CH not only receives data but also separates useful data from useless ones), we should be interested in constructing such automata or studying their properties.

We call the concept defined below quasi-automaton even though this term is not much frequent (we do this to be consistent with some earlier papers on hyperstructure theory). In fact, we could speak of semiautomata or deterministic finite automata (the below mentioned paper Chvalina and Chvalinová [25] uses a general term automaton; however, notice that [25], p. 107, plain text, defines automaton in the way of Definition 4, which is a definition adopted by the authors of [25] in later years). For an overall discussion of the concepts and the reasons for our choice of the name see Novák et al. [30]. For some further reading and applications see also Hošková et al. [31–33].

Definition 4. *By a quasi–automaton we mean a structure* $\mathbb{A} = (I, S, \delta)$ *such that* $I \neq \emptyset$ *is a monoid,* $S \neq \emptyset$ *and* $\delta : I \times S \rightarrow S$ *satisfies the following condition:*

1. *There exists an element* $e \in I$ *such that* $\delta(e, s) = s$ *for any state* $s \in S$*;*
2. $\delta(y, \delta(x, s)) = \delta(xy, s)$ *for any pair* $x, y \in I$ *and any state* $s \in S$*.*

The set I *is called the input set or input alphabet, the set* S *is called the state set and the mapping* δ *is called next-state or transition function. Condition 2 is called GMAC (Generalized Mixed Associativity Condition).*

In [25], Chvalina and Chvalinová defined what they called a state hypergroup of an automaton. This is in fact a state set with a special hyperoperation, defined by means of the transition function. In this way, the concept of a state hypergroup is fixed to the automata theory. However, the way of defining this concept is a parallel to the concept of quasi-order hypergroups, which means that state hypergroups of quasi-automata are quasi-order hypergroups. The fact that the below defined $(S, \circ)$ is a hypergroup, or rather quasi-order hypergroup, (hence the name state hypergroup) was proved in [25]. (Notice that in [25] I and S are swapped.)

Definition 5. *Let* $\mathbb{A} = (I, S, \delta)$ *be an automaton. We define a binary hyperoperation "*$\circ$*" on the state set* S *by:*

$$s \circ t = \delta(I^*, s) \cup \delta(I^*, t) \tag{3}$$

for any pair of states $s, t \in S$*, where* A^* *is a free monoid of words over the (non-empty) alphabet* A*. The hyperstructure* $(S, \circ)$ *is called state hypergroup of the automaton* $\mathbb{A}$*.*

Some properties of automata following from properties of its state hypergroup are proved in [25]. This includes the properties of being connected or separated.

Definition 6. *Let* $\mathbb{A} = (I, S, \delta)$ *be a quasi-automaton. A quasi-automaton* $\mathbb{B} = (I, S_1, \delta_1)$ *such that* $S_1 \subseteq S$ *and* δ_1 *is a restriction of* δ *on* $I \times S_1$ *and* $\delta(a, s) \in S_1$ *for any state* $s \in S_1$ *and any word* $a \in I^*$*, is called a sub quasi-automaton of* $\mathbb{A}$*. A sub quasi-automaton* $\mathbb{B} = (I, S_1, \delta_1)$ *of a quasi-automaton* $\mathbb{A} = (I, S, \delta)$ *is called separated if* $\delta(S \setminus S_1, I^*) \cap S_1 = \emptyset$*. A quasi-automaton is called connected if it does not posses any separated proper subautomaton. A quasi-automaton* $\mathbb{A} = (I, S, \delta)$ *is called strongly connected if for any states* $s, t \in S$ *there exists a word* $a \in I^*$ *such that* $\delta(a, s) = t$*.*

If in quasi–automata we suppose that the input set I is a semihypergroup instead of a free monoid, we arrive at the concept of a quasi–multiautomaton. When defining this concept, caution must be exercised when adjusting the conditions imposed on the transition function δ as on the left-hand side of condition 2 we get a state while on the right-hand side we get *a set of states*. However, in the dichotomy *deterministic* — *nondeterministic*, quasi–multiautomata still are deterministic because the range of δ is S. The difference between the transition function of a quasi-automaton and the transition function of a quasi-multiautomaton is that in quasi-automata the state achieved by applying y in a state, which is the result of application of x in s, is the same as the state achieved by applying xy in s, while condition (4) says that it is one of the many states achievable by applying any command from $x * y$ in state s.

Definition 7. *A quasi–multiautomaton is a triad $\mathbb{A} = (I, S, \delta)$, where $(I, *)$ is a semihypergroup, S is a non–empty set, and $\delta : I \times S \to S$ is a transition map satisfying the condition:*

$$\delta(b, \delta(a, s)) \in \delta(a * b, s) \text{ for all } a, b \in I, s \in S. \tag{4}$$

The hyperstructure $(I, *)$ is called the input semihypergroup of the quasi–multiautomaton $\mathbb{A}$ (I alone is called the input set or input alphabet), the set S is called the state set of the quasi–multiautomaton $\mathbb{A}$, and δ is called next-state or transition function. Elements of the set S are called states, elements of the set I are called input symbols.

Further on, we will make use of the above mentioned concepts to model the process of data aggregation. Notice that in Novák et al. [30], Cartesian composition of automata resulting in a quasi-multiautomaton is used to describe a task from collective robotics. Moreover, in Chvalina et al. [18], the issue of state sets and input sets having the form of vectors and matrices (of both numbers and special classes of functions) is discussed in the context of quasi-multiautomata.

In Figure 2 we can see that the elements of the UWSN are divided into several tiers. Also, the nodes are grouped into clusters. The process of data aggregation happens as follows: First, data is collected in cluster heads and then transmitted between cluster heads towards the surface, i.e., "upwards". Obviously, we can only transmit the amount of data that the capacity of available memory allows. Suppose that clusters cover areas of more or less the same size, i.e., it does not matter how many nodes there are in respective clusters. Now, regard a set of vectors:

$$S_v = \{\vec{v} = (v_1, v_2, \ldots, v_n) \mid v_i \in \langle 0; 1\rangle; i \in \{1, \ldots, n\}\} \tag{5}$$

of such a number of components that corresponds to the number of tiers (with index 1 meaning surface and index n meaning seabed or the deepest tier). The components v_i carry information about how much total memory all cluster heads at a given tier has been used. In other words, $v_2 = 0.6$ means that at the second tier 60% memory capacity has been used, regardless of whether this 60% means that every cluster head at this level has 40% free capacity or whether 6 out of 10 cluster heads already have no available memory while 4 are 100% free.

The process of data aggregation starts with $\vec{v} = (0, \ldots, 0)$, i.e., at the moment when all cluster heads have empty memory. Since, at first data are collected within clusters, $\vec{v}$ immediately becomes non-zero. The process of communication between cluster heads is described by the change of $\vec{v}$ by means of multiplying $\vec{v}$ by a square matrix of real numbers:

$$I_M = \left\{ \mathbf{A}_k = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix} \mid a_{ij} \in \langle 0; 1\rangle \text{ for } i \geq j, a_{ij} = 0 \text{ for } i < j; i, j \in \{1, \ldots, n\}; \|\mathbf{A}_k\|_1 \leq 1 \right\},$$

where $\|\mathbf{A}_k\|_1$ is the column norm of $\mathbf{A}_k$. In other words, I_M is a subset of the set of upper triangular matrices, i.e., we can also write:

$$I_M = \left\{ \mathbf{A}_k = \begin{bmatrix} a_{11} & 0 & 0 & \ldots & 0 \\ a_{21} & a_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix} \mid \|\mathbf{A}_k\|_1 \leq 1 \right\} \tag{6}$$

Now, these two sets S_v and I_M will be linked with a transition function δ by:

$$\delta(\mathbf{A}, \vec{v}) = \vec{v} \cdot \mathbf{A} \tag{7}$$

for all $\mathbf{A} \in I_M$ and all $\vec{v} \in S_v$. If we regard the usual matrix multiplication (only swapped), i.e., $\mathbf{A} \odot \mathbf{B} = \mathbf{B} \cdot \mathbf{A}$ for all $\mathbf{A}, \mathbf{B} \in I_M$, then $(I_M, \odot)$ is a monoid such that the free monoid $I_M^* = I_M$. Thus we can regard the triple (I_M, S_v, δ) and study whether it is a quasi-automaton. In our context, the operation of creating words from input symbols of our alphabet I will be matrix multiplication, i.e., a word will be a product of matrices, i.e., again a matrix.

Theorem 1. *The triple* (I_M, S_v, δ) *is a quasi-automaton.*

Proof. The identity matrix E_n is the neutral element of I_M. Property 1 of Definition 4 holds trivially. Verification of Property 2 is also straightforward:

$$\delta(\mathbf{A}, \delta(\mathbf{B}, \vec{v})) = \delta(\mathbf{A}, \vec{v} \cdot \mathbf{B}) = (\vec{v} \cdot \mathbf{B}) \cdot \mathbf{A} = \vec{v} \cdot \mathbf{B} \cdot \mathbf{A} = \delta(\mathbf{B} \cdot \mathbf{A}, \vec{v}) = \delta(\mathbf{A} \odot \mathbf{B}, \vec{v}).$$

□

The set I_M consists of matrices such that the column norm $||\mathbf{A}||_1$ is at most one. In the following Remark, we show that without this condition the set I_M would not be closed with respect to "$\odot$".

Remark 1. *Suppose two matrices* $\mathbf{A}, \mathbf{B} \in I_M$. *If we denote:*

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & 0 & 0 & \dots & 0 \\ b_{21} & b_{22} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n1} & b_{n2} & b_{n3} & \dots & b_{nn} \end{bmatrix},$$

then,

$$\mathbf{B} \odot \mathbf{A} = \mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} \sum\limits_{i=1}^{1} a_{1i} b_{i1} & 0 & 0 & \dots & 0 \\ \sum\limits_{i=1}^{2} a_{2i} b_{i2} & \sum\limits_{i=2}^{2} a_{2i} b_{i2} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum\limits_{i=1}^{n} a_{ni} b_{in} & \sum\limits_{i=2}^{n} a_{ni} b_{in} & \sum\limits_{i=3}^{n} a_{ni} b_{in} & \dots & \sum\limits_{i=n}^{n} a_{ni} b_{in} \end{bmatrix}$$

and it is obvious that the column norm $\|\mathbf{B} \odot \mathbf{A}\|_1$ *will not exceed 1, i.e.,* $\mathbf{B} \odot \mathbf{A} \in I_M$. *Indeed, suppose that* $\mathbf{A}$ *is an all-ones matrix upper triangular matrix (which, of course violates the condition that* $\|\mathbf{A}\|_1 \leq 1$*). Then all the sums in* $\mathbf{B} \odot \mathbf{A}$ *reduce to* $\sum\limits_{i=j}^{n} b_{ij}$ *which are (due to the fact that* $\|\mathbf{B}\|_1 \leq 1$*) smaller than* 1. *If moreover* $\|\mathbf{A}\|_1 \leq 1$, *none of the sums becomes greater. Of course, in* I_M *we could have used the row norm instead of column one with the same result.*

Example 1. *Regard sensor nodes deployed under water, which are divided into four tiers with tier 1 being* $0 - 25$ *m, tier 2 being* $25 - 50$ *m, tier 3 being* $50 - 75$ *m, and tier 4 being* $75 - 100$ *m under water. Every tier has an arbitrary number of clusters, i.e., an arbitrary number of cluster heads, each with the same memory capacity. Then by vector e.g.,* $\vec{v} = (0.3; 0.1; 0; 0.5)$, *we describe such a state of the system that, at a certain* *point in time,* 30% *data has been collected (or rather,* 30% *memory capacity has been used) from tier 1 with the numbers being* 10% *from tier 2,* 0% *from tier 3, and* 50% *from tier 4. Now regard a matrix:*

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in I_M,$$

which we apply on $\vec{v}$. Then $\delta(\mathbf{A}, \vec{v}) = (0.1; 0.1; 0.5; 0)$, which describes the new state of the system. Of course, the first 30% *of data (represented by the first component of $\vec{v}$) did not get lost, which we can regard as output, i.e., to each quasi-automaton we can assign output, meaning that data from the upmost tier 1 have been processed, sent out of the system. Furthermore, in our example, data from tier 2 were transferred to tier 1 (and left in tier 2, i.e., in this case we have a backup copy). Also, data from tier 4 were transferred to tier 3, which had been empty, and simultaneously deleted in tier 4. It can be easily observed that the issue of suitable construction of matrices used for quasi-automata inputs is a topic for further research which will be closely linked to various aspects of optimization theory.*

When describing the state hypergroup of (I_M, S_v, δ), we must first of all decide whether or not $(S_v, \circ)$ defined by Equation (3) is trivial. However, this task is rather simple.

Lemma 6. *The state hypergroup $(S_v, \circ)$ of (I_M, S_v, δ) is not trivial, i.e., there exist states $s, t \in S_v$ such that $\emptyset \neq \delta(I_M^*, s) \cup \delta(I_M^*, t) \neq S_v$.*

Proof. The transition function δ is defined on (I_M, S_v, δ) by $\delta(\mathbf{A}, \vec{v}) = \vec{v}\mathbf{A}$ for all $\mathbf{A} \in I_M$ and all $\vec{v} \in S_v$. Moreover, by Equation (5), components of vectors from S_v are real numbers from interval $\langle 0; 1 \rangle$, and by (6) matrices from I_M are upper triangular with entries taken from the same interval. Finally, in a text preceding Theorem 1 we have already mentioned that $I_M^* = I_M$. Thus, for all $s, t \in I_M$ and all $\mathbf{A} \in I_M$, we have that:

$$\delta(\mathbf{A}, \vec{s}) = \vec{s}\mathbf{A} = (s_1, s_2, \ldots, s_n) \begin{bmatrix} a_{11} & 0 & 0 & \ldots & 0 \\ a_{21} & a_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix} = \left(\sum_{i=1}^{n} s_i a_{i1}, \sum_{i=2}^{n} s_i a_{i2}, \ldots \sum_{i=n-1}^{n} s_i a_{in-1}, s_n a_{nn} \right).$$

Now suppose that $s_n = 0.1$. Obviously in this case there cannot be $\delta(\mathbf{A}, \vec{s}) = S_v$ because we are not able to find such a matrix $\mathbf{A} \in I_M^* = I_M$ that $s_n a_{nn} = 0.9$ as this would require $a_{nn} = 9$ which is out of the interval $\langle 0; 1 \rangle$ in which all entries of $\mathbf{A}$ should be. Naturally, there are infinitely many of these choices. The fact that $\delta(I_M^*, s) \cup \delta(I_M^*, t) \neq \emptyset$ is obvious. □

Regard now the set $\hat{I}_M$ which will be different from I_M because the condition of the column norm will be dropped and we will assume that $b_{ii} \geq b_{ij}$ for all $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, \ldots, i\}$, i.e., that each diagonal element will be greater than all other elements in the given row.

On $\hat{I}_M$ we define a hyperoperation "$*$" by:

$$\mathbf{A} * \mathbf{B} = \left\{ \begin{bmatrix} c_{11} & 0 & 0 & \ldots & 0 \\ c_{21} & c_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \ldots & c_{nn} \end{bmatrix} \mid c_{ii} \in \langle a_{ii} \cdot b_{ii}; 1 \rangle \text{ and } c_{ij} \in \langle a_{ij} \cdot b_{ij}; a_{ii} \cdot b_{ii} \rangle \text{ for } i \neq j \right\} \quad (8)$$

for all $\mathbf{A}, \mathbf{B} \in \hat{I}_M$, where "$\cdot$" is the usual multiplication of real numbers and c_{ij} belongs to a closed interval bounded by the product $a_{ij} \cdot b_{ij}$ and either 1 or the product of the corresponding diagonal elements.

Remark 2. *Notice that by removing the assumption regarding the column norm we are making our model more real-life because we regard memory losses and overflows. Indeed, suppose we have a vector e.g.,* $\vec{v} = (0.8; 0.7)$ *and a matrix e.g.,* $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$*. Then* $\delta(\mathbf{A}, \vec{v}) = \vec{v}\mathbf{A} = (1.5; 0.7)$*, which (since vector components are taken from the interval* $\langle 0; 1 \rangle$ *because they describe percentages of memory capacity) must be reduced to* $(1; 0.7)$*.*

Theorem 2. *The pair* $(\hat{I}_M, *)$ *is a commutative hypergroup.*

Proof. Commutativity of the hyperoperation "$*$" is obvious because we work with entry-wise multiplication of real numbers.

For the test of associativity, i.e., $\mathbf{A} * (\mathbf{B} * \mathbf{C}) = (\mathbf{A} * \mathbf{B}) * \mathbf{C}$ for all triples $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \hat{I}_M$, recall that we will be comparing sets of matrices of the same dimension, elements of which fall within intervals defined by Equation (8). As far as diagonal elements are concerned, there is obviously no problem because we make use of the multiplication of real numbers. For the non-zero, non-diagonal elements the lower bounds of the intervals are the products of their smallest elements, i.e., $a_{ij}(b_{ij}c_{ij}) = (a_{ij}b_{ij})c_{ij}$, while the upper bounds of the intervals are products of their greatest possible, i.e., diagonal, elements i.e., $a_{ii}(b_{ii}c_{ii}) = (a_{ii}b_{ii})c_{ii}$. Thus, in all cases, associativity of the hyperoperation follows from the associativity of (positive) real numbers (smaller than 1).

Reproductive axiom is also obviously valid because since the null matrix $\mathbf{0}$ is an element of $\hat{I}_M$ and $\mathbf{A} * \mathbf{0} = \hat{I}_M$. □

Notice that in the following theorem, $\delta(\mathbf{A}, \vec{v})$ is the usual vector and matrix multiplication with the additional condition that no diagonal component is allowed to exceed 1. For its motivation see Remark 2.

Theorem 3. *The triple* $(\hat{I}_M, S_v, \delta)$ *is a quasi-multiautomaton, where, for all* $\mathbf{A} \in \hat{I}_M$ *and all* $\vec{v} \in S_v$*,*

$$\delta(\mathbf{A}, \vec{v}) = \left(\sum_{i=1}^{n} v_i a_{i1}, \ldots, \sum_{i=1}^{n} v_i a_{in} \right), \tag{9}$$

where in case that $\sum_{i=1}^{n} v_i a_{ik} > 1$ *for some* $k \in \{1, 2, \ldots, n\}$*, we set, by default,* $\sum_{i=1}^{n} v_i a_{ik} = 1$.

Proof. Thanks to Theorem 2, we shall verify condition Equation (4) only. The left-hand side of the condition is:

$$\delta(\mathbf{A}, \delta(\mathbf{B}, \vec{v})) = \delta(\mathbf{A}, \vec{v} \cdot \mathbf{B}) = \delta\left(\mathbf{A}, \left(\sum_{i=1}^{n} v_i b_{i1}, \ldots, \sum_{i=1}^{n} v_i b_{in}\right)\right) = \left(\sum_{i=1}^{n} v_i b_{i1}, \ldots, \sum_{i=1}^{n} v_i b_{in}\right) \cdot \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix},$$

which, if we regard that **A**, **B** are upper triangular matrices, is:

$$\Big(\sum_{i=1}^{n} v_i b_{i1} a_{11} + \sum_{i=2}^{n} v_i b_{i2} a_{21} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,1} + \sum_{i=n}^{n} v_i b_{in} a_{n1},$$
$$\sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} a_{22} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,2} + \sum_{i=n}^{n} v_i b_{in} a_{n2},$$
$$\sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} 0 + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,n-1} + \sum_{i=n}^{n} v_i b_{in} a_{nn}, \ldots$$
$$\ldots, \sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} 0 + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} 0 + \sum_{i=n}^{n} v_i b_{in} a_{nn}\Big)$$

On the right-hand side of the condition we get:

$$\delta(\mathbf{A} * \mathbf{B}, \vec{v}) = \left(\left\langle \sum_{i=1}^{n} v_i a_{i1} b_{i1}; 1\right\rangle, \left\langle \sum_{i=2}^{n} v_i a_{i2} b_{i2}; 1\right\rangle, \ldots, \left\langle \sum_{i=n-1}^{n} v_i a_{in-1} b_{in-1}; 1\right\rangle, \left\langle \sum_{i=n}^{n} v_i a_{in} b_{in}; 1\right\rangle\right),$$

which is a vector of intervals upper-bounded by 1 such that their lower bounds are sums of lower bounds of respective intervals in the definition of the hyperoperation in Equation (8) after multiplication by vector $\vec{v}$ has been performed. (Notice that we work with non-negative matrix and vector entries only.) We have to show that:

$$\begin{aligned}
\sum_{i=1}^{n} v_i a_{i1} b_{i1} &\leq \sum_{i=1}^{n} v_i b_{i1} a_{11} + \sum_{i=2}^{n} v_i b_{i2} a_{21} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,1} + \sum_{i=n}^{n} v_i b_{in} a_{n1} \\
\sum_{i=2}^{n} v_i a_{i2} b_{i2} &\leq \sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} a_{22} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,2} + \sum_{i=n}^{n} v_i b_{in} a_{n2} \\
&\ldots \\
\sum_{i=n-1}^{n} v_i a_{in-1} b_{in-1} &\leq \sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} 0 + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,n-1} + \sum_{i=n}^{n} v_i b_{in} a_{nn} \\
\sum_{i=n}^{n} v_i a_{in} b_{in} &\leq \sum_{i=1}^{n} v_i b_{i1} 0 + \sum_{i=2}^{n} v_i b_{i2} 0 + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} 0 + \sum_{i=n}^{n} v_i b_{in} a_{nn}
\end{aligned}$$

i.e.,

$$\begin{aligned}
\sum_{i=1}^{n} v_i a_{i1} b_{i1} &\leq \sum_{i=1}^{n} v_i b_{i1} a_{11} + \sum_{i=2}^{n} v_i b_{i2} a_{21} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,1} + \sum_{i=n}^{n} v_i b_{in} a_{n1} \\
\sum_{i=2}^{n} v_i a_{i2} b_{i2} &\leq \sum_{i=2}^{n} v_i b_{i2} a_{22} + \ldots + \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,2} + \sum_{i=n}^{n} v_i b_{in} a_{n2} \\
&\ldots \\
\sum_{i=n-1}^{n} v_i a_{in-1} b_{in-1} &\leq \sum_{i=n-1,}^{n} v_i b_{in-1,} a_{n-1,n-1} + \sum_{i=n}^{n} v_i b_{in} a_{nn} \\
\sum_{i=n}^{n} v_i a_{in} b_{in} &\leq \sum_{i=n}^{n} v_i b_{in} a_{nn}
\end{aligned}$$

Now, if we move the sums on the left-hand side of the inequalities to the right-hand side and expand all the sums, we get for the first sum, i.e., for the first component of the vectors:

$$\begin{aligned} v_1 a_{11} b_{11} + v_2 a_{21} b_{21} + v_3 a_{31} b_{31} + \ldots + v_n a_{n1} b_{n1} \quad \leq \quad & v_1 b_{11} a_{11} + v_2 b_{21} a_{11} + \ldots + v_n b_{n1} a_{11} + \\ & + v_2 b_{22} a_{21} + v_3 b_{32} a_{21} + \ldots + v_n b_{n2} a_{21} + \\ & \vdots \\ & + v_{n-1} b_{n-1,n-1} a_{n-1,1} + v_n b_{n,n-1} a_{n-1,1} + \\ & + v_n b_{nn} a_{n1}, \end{aligned}$$

which, after we put elements v_i, $i \in \{1, 2, \ldots, n\}$ in front of the brackets, means that:

$$\begin{aligned} 0 \quad \leq \quad & v_2 (b_{21} a_{11} + b_{22} a_{21} - a_{21} b_{21}) + \\ & v_3 (b_{31} a_{11} + b_{32} a_{21} + b_{33} a_{31} - a_{31} b_{31}) + \\ & \vdots \\ & v_{n-1} (b_{n-1,1} a_{11} + b_{n-1,2} a_{21} + \ldots + b_{n-1,n-1} a_{n-1,1} - a_{n-1,1} b_{n-1,1}) + \\ & v_n (b_{n1} a_{11} + b_{n2} a_{21} + \ldots b_{nn} a_{n1} - a_{n1} b_{n1}) \end{aligned}$$

which, since all numbers are non-negative, will be greater than zero if $b_{ii} \geq b_{i1}$ for all $i \in \{2, 3, \ldots n\}$. In a completely analogous manner we get for the second sum, i.e., for the second component of the vectors:

$$\begin{aligned} v_2 a_{22} b_{22} + v_3 a_{32} b_{32} + \ldots + v_n a_{n2} b_{n2} \quad \leq \quad & v_2 b_{22} a_{22} + v_3 b_{32} a_{22} + \ldots + v_n b_{n2} a_{22} + \\ & + v_3 b_{33} a_{32} + v_4 b_{43} a_{32} + \ldots + v_n b_{n3} a_{32} \\ & \vdots \\ & + v_{n-1} b_{n-1,n-1} a_{n-1,2} + v_n b_{n,n-1} a_{n-1,2} + \\ & + v_n b_{nn} a_{n2}, \end{aligned}$$

which will give us condition that $b_{ii} \geq b_{i2}$ for all $i \in \{3, 4, \ldots n\}$, etc. until the $(n-1)^{th}$ component will result in condition $b_{ii} \geq b_{i,n-1}$ for $i = n$; and the first and the last condition $b_{ii} \geq b_{ii}$ for all $i \in \{1, n\}$ hold trivially.

Thus, altogether we get that $b_{ii} \geq b_{ij}$ for all $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, \ldots, i\}$. In other words, each diagonal element must be greater than all other elements in the given row, which is exactly what we suppose $\hat{I}_M$ to be. □

Remark 3. *If we want to find out whether we are able to reach any state from any other state with the help of our transition functions (i.e., test strong conectedness), we must take into account that the fact that we regard diagonal matrices means that for known vectors* $\vec{u}, \vec{v}$ *and an unknown matrix* $\mathbf{A}$ *the equation* $\vec{u}\mathbf{A} = \vec{v}$ *results in a linear system in the row echelon form and the task of finding the unknown matrix* $\mathbf{A}$ *is equivalent to finding its solution. Of course, we have to consider the fact that* $\vec{v}$ *might have components being zero, which will affect the solvability of the linear system.*

Example 2. *The hyperoperation on the input set, i.e., the fact that we work with hypergroups, gives us a better tool for controlling the process of data aggregation. Suppose that we have the same vector as in Example 1, i.e.,* $\vec{v} = (0.3; 0.1; 0; 0.5)$*. Furthermore, assume the same context as in Example 1. Let the matrix be:*

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.9 & 1 & 0 & 0 \\ 0.2 & 1 & 1 & 0 \\ 0.01 & 0.3 & 1 & 0 \end{bmatrix} \in \hat{I}_M.$$

Suppose that in our quasi-multiautomaton this matrix is applied on vector $\vec{v}$*, i.e.*

$$\delta(\mathbf{A}, \vec{v}) = (0.3; 0.1; 0; 0.5) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.9 & 1 & 0 & 0 \\ 0.2 & 1 & 1 & 0 \\ 0.01 & 0.3 & 1 & 0 \end{bmatrix}.$$

Now, focus on the multiplication of $\vec{v}$ *and the first column of* $\mathbf{A}$*. We get that:*

$$(0.3 \cdot 1 + 0.1 \cdot 0.9 + 0 \cdot 0.2 + 0.5 \cdot 0.01; a; b; c) = (0.395; a; b; c),$$

which means that the capacity of tier 1 is filled by 39.5% *in such a way that the original* 30% *was accompanied by further* 9.5% *from tier 2 (which had originally been filled up by* 10%*, i.e., we have a* 1% *transmission loss),* 0% *from tier 3, and* 0.5% *from tier 4. In this case it is obvious that the transmission from tier 4 is extremely difficult with a substantial data loss. Theoretically, we could model the process of data aggregation with matrices with all-one columns. However, this would not be a real-life case. When looking for an optimal transition input (i.e., an optimal matrix), we could regard matrices such as those in Equation (8) with increasing entries. The definition of the hyperoperation (8) means that from a certain moment on, the memory capacity could be filled to maximum and data will start being lost, i.e., not stored in cluster heads on upper tiers anymore. However, this depends also on the initial state of the system, i.e., on the initial vector. In the case of our vector,* $\vec{v}$*, even matrix:*

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

i.e., a matrix with maximal possible entries will secure successful data aggregation because vector $\vec{v}$ *suggests that the process started when all tiers had enough free memory capacity in all cluster heads without any risk of memory overflow.*

4. Conclusions and Future Work

In Remark 3, we saw in a general case a strong connectedness of our quasi(-multi) automata was not secured. Moreover, in Lemma 6, we observed that the state hypergroup of the quasi-automaton (I_M, S_v, δ) was not trivial. These two facts are motivations for future research into properties of quasi-(multi)automata, especially in their interpretation of our real-life problem, i.e., UWSN design and data aggregation. Furthermore, the range of lemmas included in Section 2 provides a good starting point for linking our context to the already established results of the algebraic hyperstructure theory. Finally, the issue of optimal choice of input matrices is another line of possible research.

Author Contributions: Investigation, M.N., Š.K. and K.O.; Methodology, M.N., Š.K. and K.O.; Supervision, M.N., Š.K. and K.O.; Writing—original draft, M.N., Š.K. and K.O.; Writing—review & editing, M.N., Š.K. and K.O.

Funding: The first author was supported by the FEKT-S-17-4225 grant of Brno University of Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Novák, M.; Ovaliadis, K.; Křehlík, Š. A hyperstructure model of Underwater Wireless Sensor Network (UWSN) design. In Proceedings of International Conference on Numerical Analysis and Applied Mathematics (ICNAAM 2017), Thessaloniki, Greece, 25–30 September 2017.
2. Wahid, A.; Dongkyun, K. Analyzing Routing Protocols for Underwater Wireless Sensor Networks. *IJCNIS* **2010**, *2*, 253–261.
3. Domingo, M.C.; Prior, R. A distributed clustering scheme for underwater wireless sensor networks. In Proceedings of the 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, Athens, Greece, 3–7 September 2007.
4. Ayaz M. et al. A survey on routing techniques in underwater wireless sensor networks. *J. Netw. Comput. Appl.* **2011**, *34*, 1908–1927. [CrossRef]
5. Ovaliadis, K.; Savage, N. Cluster protocols in underwater sensor networks: A research review. *J. Eng. Sci. Technol. Rev.* **2014**, *7*, 171–175. [CrossRef]
6. Rault, T.; Abdelmadjid, B.; Yacine, C. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122. [CrossRef]
7. Abougamila, S.; Elmorsy, M.; Elmallah, E.S. A graph theoretic approach to localization under uncertainty. In Proceedings of the 2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, 20–24 May 2018.
8. Domingo, M.C. Optimal placement zones of wireless nodes in Underwater Wireless Sensor Networks with shadow zones, In Proceedings of the 2009 2nd IFIP Wireless Days (WD), Paris, France, 15–17 December 2009.
9. Jiang, P.; Wang, X.; Jiang, L. Node deployment algorithm based on connected tree for underwater sensor networks. *Sensors* **2015**, *15*, 16763–16785. [CrossRef]
10. Corsini, P.; Leoreanu, V. *Applications of Hyperstructure Theory*. Kluwer Academic Publishers: Dordrecht, The Netherlands, 2003.
11. Comer, S. D. Combinatorial aspects of relation. *Algebr. Universalis* **1984**, *18*, 77–94. [CrossRef]
12. Corsini, P. Binary relations and hypergroupoids. *Ital. J. Pure Appl. Math.* **2000**, *7*, 11–18.
13. Corsini, P. Hyperstructures associated with ordered sets. *Bull. Greek Math. Soc.* **2003**, *48*, 7–18.
14. Cristea, I.; Ştefănescu, M. Hypergroups and n-ary relations. *European J. Combin.* **2010**, *31*, 780–789. [CrossRef]
15. De Salvo, M.; Lo Faro, G. Hypergroups and binary relations. *Multi. Val. Logic* **2002**, *8*, 645–657.
16. Pickett, H. E. Homomorphisms and subalgebras of multialgebras. *Pac. J. Math.* **1967**, *21*, 327–342. [CrossRef]
17. Chvalina, J. *Functional Graphs, Quasi-ordered Sets and Commutative Hypergroups*; Masaryk University: Brno, Czech Republic, 1995. (in Czech)
18. Chvalina, J.; Křehlík, Š.; Novák, M. Cartesian composition and the problem of generalising the MAC condition to quasi-multiautomata. *An. Şt. Univ. Ovidius Constanţa* **2016**, *24*, 79–100.
19. Křehlík, Š.; Novák, M. From lattices to H_v–matrices. *An. Şt. Univ. Ovidius Constanţa* **2016**, *24*, 209–222. [CrossRef]
20. Novák, M. Some basic properties of *EL*–hyperstructures. *Eur. J. Combin.* **2013**, *34*, 446–459. [CrossRef]
21. Novák, M.; Cristea, I. Composition in *EL*–hyperstructures. *Hacet. J. Math. Stat.* **2019**, *48*, 45–58. [CrossRef]
22. Novák, M., Křehlík, Š. *EL*–hyperstructures revisited. *Soft Comput.* **2018**, *22*, 7269–7280. [CrossRef]
23. Novák, M.; Křehlík, Š.; Cristea, I. Cyclicity in *EL*–hypergroups. *Symmetry* **2018**, *10*, 611. [CrossRef]
24. Chvalina, J. Commutative hypergroups in the sense of Marty and ordered sets, In Proceedings of the Summer School on General Algebra and Ordered Sets, Olomouc, Czech Republic, 12–15 August 1994; pp. 19–30.
25. Chvalina, J.; Chvalinová, L. State hypergroups of automata. *Acta Math. et Inform. Univ. Ostraviensis* **1996**, *4*, 105–120.
26. Comer, S.D. Some problems on hypergroups. In *Algebraic Hyperstructures and Applications*, Vougiouklis, T., Ed.; World Scientific Publishing: Singapore, 1991; pp. 67–74.
27. Massouros, G.G.; Mittas, J. D. Languages, Automata and Hypercompositional Structures. In *Algebraic Hyperstructures and Applications*, Vougiouklis, T., Ed.; World Scientific Publishing: Singapore, 1991; pp. 137–147.
28. Heidari, D.; Davvaz, B. On ordered hyperstructures. *UPB Sci. Bull. Ser. A* **2011**, *73*, 85–96.
29. Li, N.; Martínez, J.F.; Meneses Chaus, J.; Eckert, M. A survey on underwater acoustic sensor network routing protocols. *Sensors* **2016**, *16*, 414. [CrossRef]

30. Novák, M.; Křehlík, Š.; Staněk, D. *n*–ary Cartesian composition of automata. *Soft Comput.* **2019**. [CrossRef]
31. Hošková, Š. Discrete transformation hypergroups. In Proceedings of 4th International Conference Aplimat, Bratislava, Slovakia, 1–4 February 2005; pp. 275–279.
32. Hošková, Š; Chvalina, J. A survey of investigations of the Brno research group in the hyperstructure theory since the last AHA Congress. In Proceedings of the AHA 2008: 10th International Congress-Algebraic Hyperstructures And Applications, Brno, Czech Republic, 3–9 September 2008.
33. Hošková, Š.; Chvalina, J.; Račková, P. Hypergroups of integral operators in connections with transformation structures. *AiMT* **2006**, *1*, 105–117.