

DOI: 10.20535/kpi-sn.2019.1.157245

UDC 512.6:004.62

I.A. Dychka, Ye.S. Sulema*

Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

*corresponding author: sulema@pzks.fpm.kpi.ua

ORDERING OPERATIONS IN ALGEBRAIC SYSTEM OF AGGREGATES FOR MULTI-IMAGE DATA PROCESSING

Background. The range of hardware for surveillance, monitoring, and measuring is quite wide now. It gives new opportunities for obtaining large amounts of data which describe behaviour of an object (subject, process, phenomenon) of observation. Since observation is a process fulfilling in a course of time as well as usually it is resulted in obtaining multimodal data sets, it is reasonable to represent this time-based multimodal data as a multi-image of an object of observation by employing the apparatus of the Algebraic System of Aggregates.

Objective. The objective of the research is to define ordering operations of the Algebraic System of Aggregates and to present an approach of their application for processing a multi-image of an object of observation.

Methods. The research is based on both the Algebraic System of Aggregates and the concept of multi-image which enable complex representation and processing of multimodal data. There are logical, ordering and arithmetical operations in the Algebraic System of Aggregates. Ordering operations can be used for time-based processing of real objects, subjects, processes data represented as multi-images.

Results. The ordering operations on aggregates are proposed and described. These operations allow to process data of multi-images with respect to time; in particular, they enable reordering of tuples and tuple elements.

Conclusions. The processing of complex data structures of multimodal nature, which are defined, generated, measured, or recorded in terms of time, can be fulfilled based on mathematical apparatus of the Algebraic System of Aggregates, in particular, by using the ordering operations presented in this paper. It can be useful in different areas for data processing of real-world objects (subjects, processes, phenomena), characteristics of which can be measured as multimodal data by multiple sensors and presented as aggregates and multi-images.

Keywords: algebraic system of aggregates; ordering operations; multimodal data; multi-image.

Introduction

Nowadays, the range of hardware for surveillance, monitoring, and measuring is quite wide. It gives new opportunities for obtaining large amounts of data which describe behaviour of an object, subject or process of observation. These data can be used for implementation of the Digital Twins technology [1–4].

A Digital Twin is complex data representation of the object (subject, process) which fully describes its features and can be used for prediction of its further behaviour as well as modelling its critical states.

Since observation is a process going on in course of time as well as usually it is resulted in obtaining multimodal data sets, the apparatus of the *Algebraic System of Aggregates* [5, 6] can be used to obtain a multi-image [5, 6] which can be considered as a mathematical background for a digital twin of the object (subject, process) of study.

The Algebraic System of Aggregates (ASA) is an algebraic system, a carrier of which is an arbitrary set of specific structures – *aggregates* [5, 6].

An aggregate A is a tuple of arbitrary tuples, elements of which belong to certain sets:

$$A = \llbracket M_j \mid \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N = \llbracket \{A\} \mid \langle A \rangle \rrbracket,$$

where $\{A\}$ is a tuple of sets M_j , $\langle A \rangle$ is a tuple of elements tuples $\langle a_i^j \rangle_{i=1}^{n_j}$ corresponding to the tuple of sets $(a_i^j \in M_j)$.

It means that for defining the aggregate we need to indicate the sets, elements of which belong to the aggregate within appropriate tuples, and next we need to place the tuples of the elements belonging to these sets in the same order because there is a strict relation between them: the elements of the first tuple belong to the first set, the elements of the second tuple belong to the second set, etc.

For example, if we have both systolic pressure and diastolic pressure values of a patient as well as the heart rate values we can present all the measurements as one complex data structure – the aggregate:

$$A = \llbracket M_{sp}, M_{dp}, M_p \mid \langle 185, 166, 175, 166, 171, 152 \rangle, \langle 76, 73, 74, 73, 71, 76 \rangle, \langle 74, 81, 76, 93, 97, 97, 96 \rangle \rrbracket.$$

To process aggregated data, we need to apply certain operations to them. Such operations should enable logical, ordinal, and arithmetical actions on aggregates. All these operations are proposed by the apparatus of the ASA.

Problem Statement

The objective of the research is to define ordering operations of the Algebraic System of Aggregates and to present an approach of their application for construction of a multi-image of an object (subject, process) for further analysis and processing, including its use in the Digital Twin technology.

Basic Statements of ASA

Since the ASA is an algebraic system [7–10], it consists of sets $(\mathcal{M}, \mathcal{F}, \mathcal{R})$, where \mathcal{M} is a non-empty set (carrier), elements of which are elements of the system; \mathcal{F} is a set of operations; \mathcal{R} is a set of relations.

The basic *relations* of aggregates are: Is Equal ($=$), Is Less ($<$), Is Greater ($>$), Is Equivalent (\equiv), Includes (\supset), Is Included (\subset), Precedes (\prec), Succeeds (\succ).

Operations on aggregates include *logical operations*, *ordering operations*, and *arithmetical operations*. The logical operations on aggregates are: Union, Intersection, Difference, Symmetric Difference, and Exclusive Intersection [4]. Ordering operations include: Sets Ordering, Sorting, Singling, Extraction, and Insertion. Arithmetical operations include: Elementwise Addition, Scalar Addition, Elementwise Subtraction, Scalar Subtraction, Elementwise Multiplication, Scalar Multiplication, Elementwise Division, and Scalar Division.

As it is mentioned above, the *carrier* of ASA is an arbitrary set of specific structures called aggregates. Tuple elements in an aggregate can be both crisp and fuzzy values. The notation a_i means that a value is crisp and the notation \tilde{a}_i is used for a fuzzy value. A tuple element can be undefined; its notation is $_$ or $\tilde{_}$ depending on the value type.

A tuple may be empty:

$$A = \llbracket M_1, M_2, \dots, M_N \mid \langle a_1^1, a_2^1, \dots, a_n^1 \rangle, \dots \rrbracket.$$

$$\langle \emptyset \rangle, \dots, \langle a_1^N, a_2^N, \dots, a_{n_N}^N \rangle \rrbracket.$$

An aggregate which consists only of empty tuples is called *empty*:

$$A = \llbracket M_j \mid \langle \emptyset \rangle \rrbracket_{j=1}^N.$$

An aggregate which does not include any component is called a *null-aggregate*, its notation is A_\emptyset :

$$A_\emptyset = \llbracket \emptyset \mid \langle \emptyset \rangle \rrbracket.$$

The null-aggregate plays a role of a neutral element in the algebraic system of aggregates.

The sequence order of sets and corresponding tuples in an aggregate defines how operations on the aggregate will be fulfilled. The sets in the list of sets may repeat. Such case means that the aggregate includes several tuples which consist of elements of the same type.

An *undefined aggregate* is the aggregate which includes tuples of defined sets but unknown elements values:

$$A = \llbracket M_j \mid \langle _ \rangle \rrbracket_{j=1}^N.$$

The undefined aggregate defers from the empty aggregate where tuples also belong to defined sets, but the tuples are empty. The practical value of the notion of an undefined aggregate is that it allows us to predefine aggregates when some data sequences are not obtained yet (e.g. we know from which sensor we will receive data, but the sensor is currently off and, thus, data are still unavailable).

The aggregate *length* is a quantity of tuples in it. The corresponding notation is $|A|$. The length of the null-aggregate is equal to zero: $|A_\emptyset| = 0$. The tuple *length* is a quantity of elements in it. The notation is $|\bar{a}|$. A *cumulative length* $\|A\|$ of the aggregate A is a sum of the lengths of the aggregate tuples. The cumulative length of the null-aggregate is equal to zero: $\|A_\emptyset\| = 0$.

Aggregates can be *compatible*, *quasi-compatible* or *incompatible*. Aggregates A_1 and A_2 are compatible ($A_1 \doteq A_2$) if they have equal lengths and both the type and sequence order of these aggregates are the same. Aggregates A_1 and A_2 are quasi-compatible ($A_1 \doteq A_2$) if the type and sequence order of these aggregates coincide partly. There is no requirement of the equality of aggregate

gates lengths in this case. Otherwise, aggregates A_1 and A_2 are incompatible ($A_1 \dot{=} A_2$).

Besides, aggregates A_1 and A_2 can be *hiddenly compatible*, $A_1(\dot{=} A_2)$, if $A_1 \dot{=} A_2$ or $A_1 \dot{=} A_2$ but at the same time $\{A_1\} \equiv \{A_2\}$ what means that both aggregates have the same set of sets but the order of these sets differs. For example, the aggregates D_1 and D_2 defined below are hiddenly compatible:

$$D_1 = \llbracket R, N \mid \langle 12.5, 46.9, 11.7 \rangle, \langle 31, 18, 14, 5 \rangle \rrbracket,$$

$$D_2 = \llbracket N, R \mid \langle 0, 1, 2, 3, 4, 5 \rangle, \langle 643.90, 214.81, 301.14 \rangle \rrbracket.$$

Hidden compatibility is important for the Sets Ordering operation what is discussed in the next section.

Ordering Operations in ASA

The ordering operations have a specific importance in the ASA because they allow us to compose and to operate with complex data structures called *multi-images* [5, 6]. A multi-image is a complex representation of multiple data sets describing an object (subject, process) of observation which are obtained (measured, generated) in the course of time. In mathematical sense, the multi-image is an aggregate, the first data tuple of which is a non-empty tuple of time values. These values can be natural numbers or values of any other type which can be used for evident and monosemantic representation of time. Thus, the multi-image can be defined in the following way:

$$I = \llbracket T, M_1, \dots, M_N \mid \langle t_1, \dots, t_\tau \rangle, \langle a_1^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^N, \dots, a_{n_N}^N \rangle \rrbracket,$$

where T is a set of time values; $\tau \geq n_i$, $i \in [1, \dots, N]$.

If there is a tuple, elements of which are constant in time, in the multi-image then the corresponding set in the tuple of sets is marked as \dot{M} and the tuple of identical elements is defined by one element which is supposed to be valid for every time value. For example:

$$\begin{aligned} I &= \llbracket T, R, R, N \mid \langle 1, 2, 3, 4, 5 \rangle, \langle 3.2, 4.7, 1.1, 5.6, 0.3 \rangle, \\ &\langle 5.11, 5.11, 5.11, 5.11, 5.11 \rangle, \langle 116, 125, 114, 73, 0 \rangle \rrbracket \\ &= \llbracket T, R, \dot{R}, N \mid \langle 1, 2, 3, 4, 5 \rangle, \langle 3.2, 4.7, 1.1, 5.6, 0.3 \rangle, \\ &\langle 5.11 \rangle, \langle 116, 125, 114, 73, 0 \rangle \rrbracket. \end{aligned}$$

Operations which enable reordering of the aggregate elements are necessary for data processing in terms of time.

Let us consider the following ordering operations of ASA:

1. Sets Ordering \models .
2. Ascending Sorting \uparrow .
3. Descending Sorting \downarrow .
4. Singling \parallel .
5. Extraction \times .
6. Insertion \otimes .

A *Sets Ordering* operation reorders an aggregate according to a template aggregate. The template aggregate can be arbitrary, undefined or empty.

Let us consider the aggregate A defined as:

$$A = \llbracket M_3, M_1, M_2, \dots, M_N \mid \langle a_{i_3}^3 \rangle_{i_3=1}^{n_3}, \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket.$$

Besides, let the template aggregate be defined as:

$$A_{tem} = \llbracket M_1, M_2, M_3, \dots, M_N \mid \langle _ \rangle \rrbracket.$$

Then the result of Sets Ordering operation on the aggregates A and A_{tem} is the aggregate A_{res} defined as:

$$A_{res} = A \models A_{tem} = \llbracket \{A_{tem}\} \mid \langle A \rangle \rrbracket = \llbracket M_1, M_2, M_3, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}, \langle a_{i_3}^3 \rangle_{i_3=1}^{n_3}, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket.$$

Let us consider two arbitrary aggregates A_1 and A_2 .

If $A_1 \dot{=} A_2$ then the result of Sets Ordering operation is an aggregate A_3 which is equal to A_1 :

$$A_3 = A_1 \models A_2 = \llbracket \{A_2\} \mid \langle A_1 \rangle \rrbracket = \llbracket \{A_1\} \mid \langle A_1 \rangle \rrbracket = A_1.$$

If $A_1 \dot{=} A_2$ and there is no hidden compatibility between A_1 and A_2 then the result of Sets Ordering operation is an empty aggregate A_3 :

$$A_3 = A_1 \models A_2 = \llbracket \{A_2\} \mid \langle \emptyset \rangle \rrbracket.$$

If $A_1 \dot{=} A_2$ and there is no hidden compatibility between A_1 and A_2 then the result of Sets Ordering operation is an aggregate A_3 :

$$A_3 = A_1 \models A_2 = \llbracket M_j^2 \mid \langle a_{i_j}^j \rangle_{i_j=1}^{n_j} \rrbracket_{j=1}^N,$$

where $\langle a_i^j \rangle_{i=1}^{n_j} \in M_j^2$, $M_j^2 \in \{A_2\}$, $\langle a_i^j \rangle_{i=1}^{n_j} \in \langle A_1 \rangle$, $\langle A_3 \rangle \neq \langle A_1 \rangle$.

If $A_1 (\doteq) A_2$ then the result of Sets Ordering operation is such aggregate A_3 that $A_3 \doteq A_2$ and $\langle A_3 \rangle \equiv \langle A_1 \rangle$:

$$A_3 = A_1 \doteq A_2 = \llbracket \{A_2\} | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N,$$

where $\langle a_i^j \rangle_{i=1}^{n_j} \in \langle A_1 \rangle$.

Let us consider sorting operations: *Ascending Sorting* and *Descending Sorting*. These operations enable reordering of all tuples according to new – sorted – elements order (ascending or descending) of a certain tuple (called a *primary tuple*) among all tuples of the aggregate.

Let $A_1 = \llbracket M_j | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$ and $\exists k$ such as $1 < k < N, k \neq 2$ and $n_1 > n_k > n_N, n_2 = n_k$. Then the result of Ascending Sorting operation of A_1 according elements of tuple \bar{a}^k is the aggregate A_2 such as:

$$A_2 = A_1 \uparrow \bar{a}^k = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, \langle a_\alpha^k, a_\beta^k, \dots, a_\omega^k, \dots, a_\alpha^N, a_\beta^N, \dots, a_\omega^N \rangle \rrbracket,$$

where $a_\alpha^k < a_\beta^k < \dots < a_\omega^k < \dots < a_\alpha^k$; $a_m^j \in \langle a_i^j \rangle_{i=1}^{n_j}, j = 1, \dots, N, m \in [\alpha, \beta, \dots, \omega], 1 \leq m \leq n$ and $n = n_k$ if $n_j \geq n_k$ or $n = n_j$ if $n_j < n_k$.

The result of Decending Sorting operation of A_1 according elements of tuple \bar{a}^k is the aggregate A_2 such as:

$$A_2 = A_1 \downarrow \bar{a}^k = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\omega^1, \dots, a_\omega^1, \dots, a_\beta^1, a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\omega^2, \dots, a_\beta^2, a_\alpha^2, \dots, \langle a_\omega^k, \dots, a_\omega^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle a_\omega^N, \dots, a_\omega^N, a_\beta^N, a_\alpha^N \rangle \rrbracket.$$

If $k = 1, k = 2$ or $k = N$ the sorting operation is fulfilled by the same principle.

If $n_1 = n_2 = \dots = n_k = \dots = n_N$ the result of Ascending sorting operation is:

$$A_2 = A_1 \uparrow \bar{a}^k = \llbracket M_j | \langle a_\alpha^j, a_\beta^j, \dots, a_\omega^j, \dots, a_\omega^j \rangle \rrbracket_{j=1}^N$$

and the result of Decending sorting operation is:

$$A_2 = A_1 \downarrow \bar{a}^k = \llbracket M_j | \langle a_\omega^j, \dots, a_\beta^j, \dots, a_\alpha^j, a_\alpha^j \rangle \rrbracket_{j=1}^N.$$

A variant of the sorting operations is the sorting operations with appending (*Ascending Sorting with Appending* and *Descending Sorting with Appending*) which allow to increase length of shorter tuples according to the length of the primary tuple by adding a value x ($x \in [\emptyset, _ , q], q \in M_j, 1 \leq j \leq N$) either to the end or to the beginning of the sorter tuples.

The result of Ascending Sorting with Appending at the end of the sorter tuples of the aggregate A_1 is the aggregate A_2 :

$$A_2 = A_1 \uparrow (\bar{a}^k, x) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, \langle a_\alpha^k, a_\beta^k, \dots, a_\omega^k, \dots, a_\alpha^N, a_\beta^N, \dots, a_\omega^N, x, \dots, x \rangle \rrbracket,$$

where $|\langle a_\alpha^N, a_\beta^N, \dots, a_\omega^N, x, \dots, x \rangle| = n_k$.

The result of Ascending Sorting with Appending at the beginning of the sorter tuples of the aggregate A_1 is the aggregate A_2 :

$$A_2 = A_1 \uparrow (x, \bar{a}^k) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^1, a_\beta^1, \dots, a_\omega^1, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, a_\alpha^2, a_\beta^2, \dots, a_\omega^2, \dots, \langle x, \dots, x, a_\alpha^N, a_\beta^N, \dots, a_\omega^N \rangle \rrbracket,$$

where $|\langle x, \dots, x, a_\alpha^N, a_\beta^N, \dots, a_\omega^N \rangle| = n_k$.

The result of Descending Sorting with Appending at the end of the sorter tuples of the aggregate A_1 is the aggregate A_2 :

$$A_2 = A_1 \downarrow (\bar{a}^k, x) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\omega^1, \dots, a_\omega^1, \dots, a_\beta^1, a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\omega^2, \dots, a_\beta^2, a_\alpha^2, \dots, \langle a_\omega^k, \dots, a_\omega^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle a_\omega^N, \dots, a_\omega^N, a_\beta^N, a_\alpha^N, x, \dots, x \rangle \rrbracket.$$

The result of Descending Sorting with Appending at the beginning of the sorter tuples of the aggregate A_1 is the aggregate A_2 :

$$A_2 = A_1 \downarrow (x, \bar{a}^k) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\omega^1, \dots, a_\omega^1, \dots, a_\beta^1, a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\omega^2, \dots, a_\beta^2, a_\alpha^2, \dots, \langle a_\omega^k, \dots, a_\omega^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle x, \dots, x, a_\omega^N, \dots, a_\beta^N, a_\alpha^N \rangle \rrbracket.$$

Let us consider *Singling* operation on the aggregate A_1 defined as follows:

$$A_1 = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_1^1, \dots, a_m^1, \dots, a_{m+p}^1, a_{m+p+1}^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^k, \dots, a_m^k, \dots, a_{m+p}^k, a_{m+p+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_1^N, \dots, a_m^N, \dots, a_{m+p}^N, a_{m+p+1}^N, \dots, a_{n_N}^N \rangle \rrbracket,$$

where $1 \leq k \leq N$, and let $\exists m_l, \forall l$ such as $a_m^k = a_{m+1}^k = \dots = a_{m+p}^k$, $1 \leq m \leq (n_k - p)$, $1 \leq p \leq n_k$.

The result of Singling operation on the aggregate A_1 by the tuple \bar{a}^k is the aggregate A_2 such as:

$$A_2 = A_1 \parallel \bar{a}^k = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_1^1, \dots, a_m^1, a_{m+p+1}^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^k, \dots, a_m^k, a_{m+p+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_1^N, \dots, a_m^N, a_{m+p+1}^N, \dots, a_{n_N}^N \rangle \rrbracket.$$

Let us consider *Extraction* operation on the aggregate A_1 which is defined as follows:

$$A_1 = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_{i_1}^k, \dots, a_{m-1}^k, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket.$$

Then the result of Extraction of the element a_m^k from the aggregate A_1 is the aggregate A_2 such as:

$$A_2 = A_1 \times a_m^k = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket.$$

The operation of *Conditional Extraction* is defined on the assumption of a given condition such as $a_m^k = d$, $a_m^k < d$ or $a_m^k > d$, $\forall d \in M_k$. For example: $A_3 = A_1 \times a_m^k \mid_{a_m^k=d}$.

Let us consider *Insertion* operation on the aggregate A_1 which is defined as follows:

$$A_1 = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket,$$

where $1 \leq m \leq N$ and let $\exists d$ such as $d \in M_k$, $1 \leq k \leq N$.

Then the result of Insertion of d to the aggregate A_1 is the aggregate A_2 or the aggregate A_3 such as:

$$A_2 = A_1 \times (d < a_m^k) = A_1 \times (d > a_{m-1}^k) \\ = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, d, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket,$$

$$A_3 = A_1 \times (d > a_m^k) = A_1 \times (d < a_{m+1}^k) \\ = \llbracket M_1, \dots, M_k, \dots, M_N \mid \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_m^k, d, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket.$$

The operation of *Conditional Insertion* is defined on the assumption of a given condition such as $d = a_m^k$, $d < a_m^k$ or $d > a_m^k$, $\forall d \in M_k$. For example:

$$A_3 = A_1 \times (d < a_m^k) \mid_{d < a_m^k, m=m_1 \dots m_2},$$

$$A_4 = A_1 \times (d > a_m^k) \mid_{d > a_m^k, m=m_1 \dots m_2}.$$

Thus, the ordering operations in the ASA enable reordering of both tuples in the aggregate and elements in tuples.

Multi-Image Processing Approach

The precondition of a multi-image construction is that obtained (generated, measured) data sequences are of different modalities and they are recorded (generated, measured) with respect to time.

While the logical operations of the ASA are used for preparing complex data representations (aggregates and multi-images) [5], the ordering operations can be used for processing of complex data structures (aggregates and multi-images).

Let us consider an example of the ASA ordering operations application for processing of complex representation of multimodal data about a patient's health status. Let these data sequences be obtained from two digital sensors: thermometer and pulsometer. Then, let the data belong to the following data sets:

$M_t = [35.0, \dots, 39.9]$ is a set of temperature values ($^{\circ}\text{C}$);

$M_p = [50, \dots, 110]$ is a set of pulse values (bpm);

$T = [1, \dots, 31]$ is a set of time values (day of a month).

We assume that some of measurements are taken simultaneously and the other measurements are taken sequentially (in different days). Every case of measuring is presented as a certain multi-image. Thus, let us consider the following multi-images:

$$I_1 = \llbracket T, M_t, M_p \mid \langle t_{i_t} \rangle_{i_t=1}^8, \langle a_{i_1}^1 \rangle_{i_1=1}^8, \langle a_{i_2}^2 \rangle_{i_2=1}^7 \rrbracket = \llbracket T, M_t, M_p \mid \langle 9, 10, 12, 13, 17, 18, 19, 20 \rangle, \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5 \rangle, \langle 74, 81, 76, 93, 97, 97, 96 \rangle \rrbracket,$$

$$I_2 = \llbracket T, M_p, M_t \mid \langle t_{i_t} \rangle_{i_t=1}^6, \langle b_{i_1}^1 \rangle_{i_1=1}^6, \langle b_{i_2}^2 \rangle_{i_2=1}^6 \rrbracket = \llbracket T, M_p, M_t \mid \langle 2, 3, 11, 14, 21, 28 \rangle, \langle 75, 76, 74, 73, 75, 75 \rangle, \langle 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle \rrbracket.$$

Let us assume that the task is to obtain a single multi-image which sums up the results of the patient's health status monitoring being fulfilled during a month.

To solve this task, at first, we need to discover the level of compatibility of these multi-images.

It is evident that $I_1 \doteq I_2$ (they both are multi-images and therefore the first tuple in both of them belongs to the set of time values). At the same time, $\{I_1\} \equiv \{I_2\}$ but $\{I_1\} \neq \{I_2\}$. Therefore, $I_1 (\doteq) I_2$. To make these multi-images fully compatible, we should apply the operation of Sets Ordering:

$$\hat{I}_2 = I_2 \models I_1 = \llbracket T, M_t, M_p \mid \langle 2, 3, 11, 14, 21, 28 \rangle, \langle 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle, \langle 75, 76, 74, 73, 75, 75 \rangle \rrbracket.$$

In I_1 , $|\langle a_{i_2}^2 \rangle_{i_2=1}^7| < |\langle t_{i_t} \rangle_{i_t=1}^8|$ (in practice, it can happen if the pulse rate was not measured at the last day of patient's observation). Let us use Insertion operation to append the tuple $\langle a_{i_2}^2 \rangle_{i_2=1}^7$ by value $x = \emptyset$:

$$\hat{I}_1 = I_1 \times (x \succ a_7^2) = \llbracket T, M_t, M_p \mid \langle t_{i_t} \rangle_{i_t=1}^8, \langle a_{i_1}^1 \rangle_{i_1=1}^8, \langle a_{i_2}^2 \rangle_{i_2=1}^8 \rrbracket = \llbracket T, M_t, M_p \mid \langle 9, 10, 12, 13, 17, 18, 19, 20 \rangle, \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5 \rangle, \langle 74, 81, 76, 93, 97, 97, 96, \emptyset \rangle \rrbracket.$$

Next, we can compose data of all multi-images into one global multi-image I_g . It can be done by using logical operations [5], in particular, Union operation. For obviousness, let us do it sequentially.

At first, let us apply Union operation to \hat{I}_1 and \hat{I}_2 . Since $\hat{I}_1 \doteq \hat{I}_2$, the resulted multi-image $I_{1,2}$ is as follows:

$$I_{1,2} = \hat{I}_1 \cup \hat{I}_2 = \llbracket T, M_t, M_p \mid \langle t_{i_t} \rangle_{i_t=1}^{14}, \langle e_{i_1}^1 \rangle_{i_1=1}^{14}, \langle e_{i_2}^2 \rangle_{i_2=1}^{14} \rrbracket = \llbracket T, M_t, M_p \mid \langle 9, 10, 12, 13, 17, 18, 19, 20, 2, 3, 11, 14, 21, 28 \rangle, \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5, 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle, \langle 74, 81, 76, 93, 97, 97, 96, \emptyset, 75, 76, 74, 73, 75, 75 \rangle \rrbracket.$$

Let us sort $I_{1,2}$ in ascending order of time values to obtain the final result for this compound data structure:

$$\hat{I}_{1,2} = I_{1,2} \uparrow \bar{t} = \llbracket T, M_t, M_p \mid \langle t_{i_t} \rangle_{i_t=1}^{14}, \langle e_{i_1}^1 \rangle_{i_1=1}^{14}, \langle e_{i_2}^2 \rangle_{i_2=1}^{14} \rrbracket = \llbracket T, M_t, M_p \mid \langle 2, 3, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 28 \rangle, \langle 36.4, 36.1, 36.5, 36.5, 36.3, 36.8, 36.6, 36.2, 36.3, 36.4, 37.0, 36.5, 36.5, 36.3 \rangle, \langle 75, 76, 74, 81, 74, 76, 93, 73, 97, 97, 96, \emptyset, 75, 75 \rangle \rrbracket.$$

Let us consider another example from the field of health care. In this example we assume that the data are received from two sources of different nature: some data sets are a result of using digital sphygmomanometer and other sets are obtained as a result of blood tests. Thus, we operate with the following data sets:

$M_{sp} = [80, \dots, 190]$ is a set of systolic pressure values (mmHg);

$M_{dp} = [55, \dots, 100]$ is a set of diastolic pressure values (mmHg);

$M_h = [132, \dots, 161]$ is set of haemoglobin values (gram/litre);

$M_{esr} = [2, \dots, 15]$ is set of erythrocyte sedimentation rate values (mm/hour);

$T = [1, \dots, 31]$ is a set of time values (day of a month).

Let us consider the following multi-images obtained from the given measurements:

$$\begin{aligned}
 I_3 &= \llbracket T, M_{sp}, M_{dp} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^4, \langle c_{i_1}^1 \rangle_{i_1=1}^4, \langle c_{i_2}^2 \rangle_{i_2=1}^4 \rrbracket \\
 &= \llbracket T, M_{sp}, M_{dp} \mid \langle 2, 9, 16, 23 \rangle, \langle 185, 166, 175, 152 \rangle, \\
 &\quad \langle 76, 73, 74, 73 \rangle \rrbracket,
 \end{aligned}$$

$$\begin{aligned}
 I_4 &= \llbracket T, M_h, M_{esr} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^4, \langle d_{i_1}^1 \rangle_{i_1=1}^4, \langle d_{i_2}^2 \rangle_{i_2=1}^4 \rrbracket \\
 &= \llbracket T, M_h, M_{esr} \mid \langle 2, 9, 16, 23 \rangle, \langle 144, 142, 145, 154 \rangle, \\
 &\quad \langle 17, 15, 15, 13 \rangle \rrbracket.
 \end{aligned}$$

It is evident that $I_3 \doteq I_4$ because $\{I_3\} \not\equiv \{I_4\}$ but both multi-images have the same type of the first tuple since they both are multi-images.

Since $I_3 \doteq I_4$, to unite these multi-images meaningfully, we need to append tuples, elements of which belong to different sets, with empty values in order to make them equal in length to the joint tuple (time values tuple) of the multi-image $I_{3,4}$.

For this purpose, we should apply Insertion operation four times to the second and the thirds tuples of both I_3 and I_4 . As a result, we obtain the following multi-images:

$$\begin{aligned}
 \hat{I}_3 &= \llbracket T, M_{sp}, M_{dp} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^4, \langle c_{i_1}^1 \rangle_{i_1=1}^8, \langle c_{i_2}^2 \rangle_{i_2=1}^8 \rrbracket = \\
 &\llbracket T, M_{sp}, M_{dp} \mid \langle 2, 9, 16, 23 \rangle, \langle 185, 166, 175, 152, \emptyset, \emptyset, \\
 &\quad \emptyset, \emptyset \rangle, \langle 76, 73, 74, 73, \emptyset, \emptyset, \emptyset, \emptyset \rangle \rrbracket,
 \end{aligned}$$

$$\begin{aligned}
 \hat{I}_4 &= \llbracket T, M_h, M_{esr} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^4, \langle d_{i_1}^1 \rangle_{i_1=1}^8, \langle d_{i_2}^2 \rangle_{i_2=1}^8 \rrbracket \\
 &= \llbracket T, M_h, M_{esr} \mid \langle 2, 9, 16, 23 \rangle, \langle 144, 142, 145, 154, \emptyset, \\
 &\quad \emptyset, \emptyset, \emptyset \rangle, \langle 17, 15, 15, 13, \emptyset, \emptyset, \emptyset, \emptyset \rangle \rrbracket.
 \end{aligned}$$

Then, we can apply Union operation to \hat{I}_3 and \hat{I}_4 :

$$\begin{aligned}
 I_{3,4} &= \hat{I}_3 \cup \hat{I}_4 = \llbracket T, M_{sp}, M_{dp}, M_h, M_{esr} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^8, \\
 &\langle f_{i_1}^1 \rangle_{i_1=1}^8, \langle f_{i_2}^2 \rangle_{i_2=1}^8, \langle f_{i_3}^3 \rangle_{i_3=1}^8, \langle f_{i_4}^4 \rangle_{i_4=1}^8 \rrbracket = \llbracket T, M_{sp}, M_{dp}, \\
 &M_h, M_{esr} \mid \langle 2, 9, 16, 23, 2, 9, 16, 23 \rangle, \langle 185, 166, 175, 152, \\
 &\quad \emptyset, \emptyset, \emptyset, \emptyset \rangle, \langle 76, 73, 74, 73, \emptyset, \emptyset, \emptyset, \emptyset \rangle, \langle 144, 142, \\
 &\quad 145, 154, \emptyset, \emptyset, \emptyset, \emptyset \rangle, \langle 17, 15, 15, 13, \emptyset, \emptyset, \emptyset, \emptyset \rangle \rrbracket.
 \end{aligned}$$

Let us sort $I_{3,4}$ in ascending order of time values:

$$\hat{I}_{3,4} = I_{3,4} \uparrow \bar{t} = \llbracket T, M_{sp}, M_{dp}, M_h, M_{esr} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^8,$$

$$\begin{aligned}
 &\langle f_{i_1}^1 \rangle_{i_1=1}^8, \langle f_{i_2}^2 \rangle_{i_2=1}^8, \langle f_{i_3}^3 \rangle_{i_3=1}^8, \langle f_{i_4}^4 \rangle_{i_4=1}^8 \rrbracket = \llbracket T, M_{sp}, M_{dp}, \\
 &M_h, M_{esr} \mid \langle 2, 2, 9, 9, 16, 16, 23, 23 \rangle, \langle 185, \emptyset, 166, \emptyset, \\
 &175, \emptyset, 152, \emptyset \rangle, \langle 76, \emptyset, 73, \emptyset, 74, \emptyset, 73, \emptyset \rangle, \langle 144, \emptyset, \\
 &142, \emptyset, 145, \emptyset, 154, \emptyset \rangle, \langle 17, \emptyset, 15, \emptyset, 15, \emptyset, 13, \emptyset \rangle \rrbracket.
 \end{aligned}$$

Since there is duplication of time values in the multi-image $\hat{I}_{3,4}$, we should apply Singling operation to it in order to obtain the resulting multi-image:

$$\begin{aligned}
 \hat{I}_{3,4}^S &= \hat{I}_{3,4} \parallel \bar{t} = \llbracket T, M_{sp}, M_{dp}, M_h, M_{esr} \mid \langle t_{i_\tau} \rangle_{i_\tau=1}^4, \\
 &\langle f_{i_1}^1 \rangle_{i_1=1}^4, \langle f_{i_2}^2 \rangle_{i_2=1}^4, \langle f_{i_3}^3 \rangle_{i_3=1}^4, \langle f_{i_4}^4 \rangle_{i_4=1}^4 \rrbracket = \llbracket T, M_{sp}, \\
 &M_{dp}, M_h, M_{esr} \mid \langle 2, 9, 16, 23 \rangle, \langle 185, 166, 175, 152 \rangle, \\
 &\langle 76, 73, 74, 73 \rangle, \langle 144, 142, 145, 154 \rangle, \langle 17, 15, 15, 13 \rangle \rrbracket.
 \end{aligned}$$

Thus, the ordering operations of the ASA allow us to process data in terms of time according to that when they were measured, generated, recorded, etc. These operations are the mathematical background for multi-images processing to be fulfilled by using a domain-specific programming language such as ASAMPL [6].

Conclusions

The processing of complex data structures of multimodal nature which are defined, generated, measured or recorded in terms of time can be fulfilled based on mathematical apparatus of the Algebraic System of Aggregates. The ASA enables multimodal data processing by using a range of operations and relations. In particular, the ordering operations presented in this paper allow to process multi-images of objects, subjects or processes to be under observation. It can be useful in different areas, including healthcare, for data processing related to real-world objects, subjects, and processes, characteristics of which can be measured as multimodal data by multiple sensors and presented as aggregates and multi-images.

The further research should be focused on development of methods and algorithms for implementation of the multi-image concept [5] which enables overall description of an object, a subject, or a process of observation carried out in the course of time. In its turn, the multi-image concept can be considered as a possible mathematical background for complex data structures processing in Digital Twins technology.

Acknowledgement

The research has been carried out with the support of the Ministry of Education and Science of Ukraine within the framework of the projects “Health^{5G} – Future eHealth powered by 5G” of

the European research and development program EUREKA and 0117U004267 “Development and investigation of medical image processing, recognition, protection, and storage methods in distributed computer systems”.

References

- [1] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”, in *Transdisciplinary Perspectives on Complex Systems*. Springer, 2017. doi: 10.1007/978-3-319-38756-7_4
- [2] E.J. Tuegel *et al.*, “Reengineering aircraft structural life prediction using a digital twin”, *Int. J. Aerospace Eng.*, vol. 2011, Article ID 154798, 2011. doi: 10.1155/2011/154798
- [3] E. Glaessgen and D. Stargel, “The digital twin paradigm for future NASA and U.S. air force vehicles”, in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Conf. Structures, Structural Dynamics and Materials*, 2012. doi: 10.2514/6.2012-1818
- [4] Th.H.-J. Uhlemann *et al.*, “The digital twin: Realizing the cyber-physical production system for industry 4.0”, in *Proc. 24th CIRP Conf. Life Cycle Engineering*, vol. 61, pp. 335–340, 2017. doi: 10.1016/j.procir.2016.11.152
- [5] I. Dychka and Ye. Sulema, “Logical operations in algebraic system of aggregates for multimodal data representation and processing”, in *KPI Sci. News*, no. 6, pp. 44–52, 2018. doi: 10.20535/1810-0546.2018.6.151546
- [6] Ye. Sulema, “ASAMPL: Programming language for mulsemmedia data processing based on algebraic system of aggregates”, in *Interactive Mobile Communication Technologies and Learning. IMCL 2017. Advances in Intelligent Systems and Computing*, vol. 725, M. Auer and T. Tsiatsos, eds. Cham, Switzerland: Springer, 2018, pp. 431–442. doi: 10.1007/978-3-319-75175-7_43
- [7] A.I. Maltsev, *Algebraic Systems*. Moscow, SU: Nauka, 1970, 392 p.
- [8] A.A. Fraenkel *et al.*, *Foundations of Set Theory*. Elsevier, 1973, 415 p.
- [9] B.A. Kulik *et al.*, *Algebraic Approach to Intellectual Processing of Data and Knowledge*. Saint Petersburg, Russia: SPbPU Publ., 2010, 235 p.
- [10] A.B. Petrovsky, *Space of Sets and Multi-Sets*. Moscow, SU: Editorial URSS, 2003, 248 p.

I.A. Дичка, Е.С. Сулема

ОПЕРАЦІЇ ВПОРЯДКУВАННЯ В АЛГЕБРИЧНІЙ СИСТЕМІ АГРЕГАТИВ ДЛЯ ОБРОБЛЕННЯ ДАНИХ МУЛЬТІОБРАЗУ

Проблематика. На сьогодні спектр апаратного забезпечення для спостереження, моніторингу та вимірювань є доволі широким. Це надає нові можливості для отримання великих наборів даних, які описують поведінку об'єкта (суб'єкта, процесу, явища), що підлягає вивченню. Оскільки спостереження є процесом, що відбувається в часі та який у багатьох випадках завершується отриманням численних наборів мультимодальних даних, доцільно подавати такі комплексні дані у вигляді мультіобразу об'єкта спостереження, що може бути виконано з використанням апарату алгебричної системи агрегатів.

Мета дослідження. Визначення операцій впорядкування алгебричної системи агрегатів та формування методик їх застосування для оброблення даних мультіобразу об'єкта спостереження.

Методика реалізації. Дослідження ґрунтується на застосуванні алгебричної системи агрегатів та концепції мультіобразу, які уможливають комплексне подання та оброблення мультимодальних даних. В алгебричній системі агрегатів визначено логічні операції, операції впорядкування та арифметичні операції над агрегатами. Операції впорядкування можуть використовуватися для розв'язання задач, що пов'язані з обробленням даних мультіобразів реальних об'єктів (суб'єктів, процесів, явищ) з урахуванням часу.

Результати дослідження. Запропоновано та описано операції впорядкування агрегатів. Ці операції дають змогу оброблювати дані мультіобразів з точки зору часу, зокрема, вони уможливають перевпорядкування кортежів та їх елементів.

Висновки. Оброблення комплексних структур даних мультимодальної природи, які визначаються, генеруються, вимірюються або записуються з урахуванням часу, може бути виконане на основі застосування апарату алгебричної системи агрегатів, зокрема операцій впорядкування, що запропоновані у цій статті. Це надає можливість розв'язання задач у різноманітних галузях, де є потреба в обробленні характеристик реальних об'єктів (суб'єктів, процесів, явищ), які можуть бути виміряні за допомогою набору даних і подані як мультимодальні дані у вигляді агрегатів та мультіобразів.

Ключові слова: алгебрична система агрегатів; операції впорядкування; мультимодальні дані; мультіобраз.

I.A. Дичка, Е.С. Сулема

ОПЕРАЦИИ УПОРЯДОЧИВАНИЯ В АЛГЕБРАИЧЕСКОЙ СИСТЕМЕ АГРЕГАТОВ ДЛЯ ОБРАБОТКИ ДАННЫХ МУЛЬТИОБРАЗА

Проблематика. На сегодняшний день спектр апаратного обеспечения для наблюдения, мониторинга и измерений является достаточно широким. Это дает новые возможности для получения больших наборов данных, которые описывают поведение объекта (субъекта, процесса, явления), подлежащего изучению. Поскольку наблюдение является процессом, который осуществляется во времени и который во многих случаях завершается получением множественных наборов мультимодальных

данных, целесообразно представлять такие комплексные данные в виде мультиобраза объекта наблюдения, что может быть выполнено с использованием аппарата алгебраической системы агрегатов.

Цель исследования. Определение операций упорядочивания алгебраической системы агрегатов и формирование методики их применения для обработки данных мультиобраза объекта наблюдения.

Методика реализации. Исследование основывается на применении алгебраической системы агрегатов и концепции мультиобраза, которые позволяют комплексное представление и обработку мультимодальных данных. В алгебраической системе агрегатов определены логические операции, операции упорядочивания и арифметические операции над агрегатами. Операции упорядочивания могут использоваться для решения задач, связанных с обработкой данных мультиобразов реальных объектов (субъектов, процессов, явлений) с учетом времени.

Результаты исследования. Предложены и описаны операции упорядочивания агрегатов. Эти операции позволяют обрабатывать данные мультиобразов с точки зрения времени, в частности, они позволяют переупорядочение кортежей и их элементов.

Выводы. Обработка комплексных структур данных мультимодальной природы, которые определяются, генерируются, измеряются или записываются с учетом времени, может быть выполнена на основе применения аппарата алгебраической системы агрегатов, в частности операций упорядочивания, предложенных в этой статье. Это дает возможность решения задач в разнообразных областях, где есть потребность в обработке характеристик реальных объектов (субъектов, процессов, явлений), которые могут быть измерены с помощью набора датчиков и представлены как мультимодальные данные в виде агрегатов и мультиобразов.

Ключевые слова: алгебраическая система агрегатов; операции упорядочивания; мультимодальные данные; мультиобраз.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
17 грудня 2018 року

Прийнята до публікації
28 лютого 2019 року