The Algebra of Signatures for Extreme Two-Uniform Hypergraphs

Evgeniya Egorova 1,†, Aleksey Mokryakov 2,*,† and Vladimir Tsurkov 1,†

1 Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 119333 Moscow, Russia; eegorova@frccsc.ru (E.E.); tsur@ccas.ru (V.T.)
2 Institute of Information Technology and Digital Transformation, Kosygin Russian State University (Technologies. Design. Art.), 119071 Moscow, Russia
* Correspondence: mokryakov-av@rguk.ru
† These authors contributed equally to this work.

Abstract: In the last decade, several characterizations have been constructed for constructions such as extreme hypergraphs. One of the most recently described features is the signature. A signature is a number that uniquely describes an extremal and allows one to efficiently store the extremal two-uniform hypergraph itself. However, for the signature, although various algorithms have been derived for transforming it into other object-characteristics such as the base, the adjacency matrix, and the vector of vertex degrees, no isolated signature union and intersection apparatus has been constructed. This allows us to build efficient algorithms based on signatures, the most compact representation of extremal two-uniform hypergraphs. The nature of the algebraic construction that can be built on a set of signatures using union and intersection operations has also been defined. It is proved that an algebra on a set of signatures with either the union or intersection operation forms a monoid; if the algebra is defined on a set of signatures with both union and intersection operations, it forms a distributive lattice.

Keywords: extremeuniform hypergraph; signature; algebra

MSC: 06D99

1. Introduction

The theory of hypergraphs was born in the middle of the last century [1], but it did not have great practical significance initially. Major researchers were mainly engaged in the theoretical description of elements of the theory [2–8], although they were looking for areas for practical application. The main advantage, as well as disadvantage, is the complete amorphousness of hypergraphs. In fact, researchers can use hypergraphs to describe any model where there are at least some relations between objects. Moreover, unlike graphs, it is possible to describe very complex relationships between groups of objects simultaneously [9–16].

The disadvantage of this approach is the lack of common approaches and principles in algorithm development. In most cases, we had to develop solutions for each specific problem with minimal possibilities for its scaling and multiplication. It should be noted that the main direction of the development of hypergraph theory was focused on the questions of its coloring [5,8] or construction of minimal coverings [17]. These problems, although generally practical, have great potential in investigations of the internal structure of hypergraphs of certain classes.

This situation quickly led to the emergence of more rigorous classes that imposed various restrictions, first of all, on hypercitesb:HyperNets, b:HyperWiFI, b:IntegralDeclare. One of the most rigorous and easily described classes was the class of uniform hypergraphs. It imposed a simple, logical, but strict restriction on the power of vertices incident
to hyperbras. Basically, one speaks of $k$-uniform hypergraphs, which require that each hyperedge is incident to exactly $k$ vertices. This approach was welcomed not only by discrete optimization specialists, but also by topologists [18], who have long used the term $k$-complex, which is a $k - 1$-uniform hypergraph by its structure. Complexes are subject to almost the same restrictions as uniform hypergraphs, although they are intended, first of all, for combining structures of lower dimension into a whole structure. Most of the researchers focused on uniform hypergraphs of specific dimensions: two [14], three [19], eight, and others. At the same time, some authors continued to study general cases of [13].

Toward the end of the century, the growth of interest in uniform hypergraphs stopped, but increased sharply again at the beginning of the next century. The main drivers were applied research, which allowed not only for the construction of algorithms for solving practical problems, but also helped in the development of theoretical approaches [20].

One of the new directions in uniform hypergraphs is related to optimization of their processing and storage, as they turned out to be in high demand in social network modeling. The speed of data processing increased as well as the volume of data involved, though the latter grew faster.

The main ways of storing uniform hypergraphs are $k$-dimensional adjacency matrices [21], incident matrices, and vectors of vertex degrees. The latter have one significant drawback at high data density: the ambiguity of recovering uniform hypergraphs from a vector of vertex degrees [22]. This problem has given rise to interest in a narrower class of uniform hypergraphs: extreme hypergraphs. Their main advantage is that there is a one-to-one correspondence between the degree vectors of vertices and the extremal hypergraphs corresponding to them. Other compact characterizations, “base” [23] and “signature” [24], were obtained for the same class of hypergraphs. The latter is true only for the two-dimensional and three-dimensional cases. The obtained characterizations have applications in new algorithms of data encryption [10,12,16], and also help in studies of the class of extreme hypergraphs [23] itself.

The idea of the signature as the most compact representation of an extremal hypergraph has been developed [25], and efficient algorithms for transformation between different representations of extremal uniform hypergraphs have been developed [19].

This paper aims to further develop signatures and proposes the construction of algebraic constructions on the set of signatures of fixed dimension. The goal is to identify specific admissible algebraic constructions built on a set of signatures using one or two operations. We will analyze algebraic structures such as semigroups, monoids, groups, rings, fields, and lattices.

The Materials and Methods section provides a well-known description of signatures and a description of operations on them. The operations are illustrated by examples, including graphs defined by signatures. Then, the basic algebraic structures and their properties are recalled.

The results of the study are presented in the subsections of the next section. At the beginning, two new descriptions of union and intersection operations closed with respect to signatures, i.e., performed without reference to adjacency matrices, are given. The execution of the described operations is also illustrated by examples.

The next subsection is devoted to determining the possibility of constructing algebraic structures such as semigroups, monoids, groups, rings, and fields. The construction of a distributive lattice on a set of signatures is considered in the final subsection.

2. Materials and Methods

First, let us recall the meaning of the notion of signature. It is an $n - 1$-bit binary non-negative integer. The signature is also a broken curve separating the domains of zeros and ones in the upper triangular part of the adjacency matrix.

Example 1. Consider a graph $G$ built on 12 vertices. Its adjacency matrix is given in Table 1, and the graph itself is shown in Figure 1. In the adjacency matrix, a continuous polyline running from
the upper right corner to the main diagonal sets the signature: each vertical segment of unit length sets 1 in the corresponding digit, and the horizontal one sets 0. In this case, the construction of the signature goes from the highest to the lowest digit.

Also, by this example, it is easy to understand why there are 1 fewer digits in the signature than rows in the adjacency matrix, or (what is the same) vertices in the described graph: the polyline ends touching the cell on the main diagonal.

![Graph described by the signature](image)

**Figure 1.** Graph described by the signature $\sigma = 01110010112$ from Example 1.

**Table 1.** Signature $\sigma = 01110010112$ for the adjacency matrix.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When working with sets, the operations intersection and union are usually used. Signatures are not sets in pure form, but they are a characteristic describing extreme graphs, which in turn are a set of edges represented on a set of vertices. Thus, we can define the operation of combining the signatures $\sigma_1$ and $\sigma_2$, of graphs $G_1$ and $G_2$, respectively, as the signature $\sigma_3$ describing the graph $G_3 = G_1 \cap G_2$. Intersection can be defined similarly.

The operations of union and intersection of signatures can also be described through binary adjacency matrices.

**Definition 1.** Let the signatures $\sigma_1$ and $\sigma_2$ be given with their corresponding binary adjacency matrices $X_1 = (x_{1ij})$ and $X_2 = (x_{2ij})$. Then, the signature $\sigma_3 = \sigma_1 \cup \sigma_2$ is defined by the binary matrix $X_3 = (x_{3ij})$, where $x_{3ij} = \max_{ij}(x_{1ij}, x_{2ij})$. The intersection of signatures is defined analogously; only the elements of the leading matrix $X_3$ are defined by the minimum: $x_{3ij} = \min_{ij}(x_{1ij}, x_{2ij})$.

Let us demonstrate this definition with the following example.
Example 2. Let the signatures \( \sigma_1 = 0111001_2 \) and \( \sigma_2 = 1100101_2 \) be given; their adjacency matrices are shown in Table 2 and graphs itself is shown in Figures 2 and 3. The results of union and intersection are shown in Table 3 (graphs in Figures 4 and 5).

![Graph described by the signature \( \sigma_1 = 0111001_2 \) from Example 2.](image1)

![Graph described by the signature \( \sigma_2 = 1100101_2 \) from Example 2.](image2)

Figure 2. Graph described by the signature \( \sigma_1 = 0111001_2 \) from Example 2.

Figure 3. Graph described by the signature \( \sigma_2 = 1100101_2 \) from Example 2.

<table>
<thead>
<tr>
<th>Table 2. Input data for Example 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

\( \sigma_1 = 0111001_2 \) \quad \sigma_2 = 1100101_2
Since the goal of the study is to determine the most complex algebraic structure that can be built on a set of signatures, we recall the hierarchy of the different structures.

**Algebraic structure**, or simply algebra, is \((X; F)\), where \(X\) is a set of elements with a set of operations \(F = \varphi_1, \ldots, \varphi_2\) defined on them, and we usually consider the case when \(X\) is closed with respect to the operations of set \(F\). If only a part of the set \(X\) is closed with respect to \(F\), then such a structure is called a *subalgebra*.

We mainly consider algebras with one and two operations. Let us first list the algebras with one binary operation, which can be denoted differently. We will define it as the sign...
If the nature of the operation is known, it is often denoted by $\oplus$ or $\otimes$ depending on its behavior: addition or multiplication.

**Definition 2.** Semigroup—an algebra $\langle X; \circ \rangle$ with associative binary operation $\circ$.

Monoid—a semigroup with unit element $e$ that satisfies the following property: $\forall x \in X \; x \circ e = e \circ x = x$.

Group—a monoid with a definite inverse element $x^{-1}$ for every element of the set $X$.

An Abelian group is a group with a commutative binary operation.

It is also customary to distinguish a number of algebras with two binary operations $\oplus$ or $\otimes$.

**Definition 3.** Ring is an algebra $\langle X; \oplus, \otimes \rangle$ where a number of properties hold:

$\langle X; \oplus, \otimes \rangle$ is an Abelian group by “union”;

$\langle X; \oplus, \otimes \rangle$ is a semigroup by intersection;

$\otimes$—distibutively by left and right multiplication.

If the corresponding conditions are met, the ring can be commutative, as well as being a commutative ring with a unit with respect to the multiplication operation.

**Definition 4.** A field is a commutative ring with a unit relative to multiplication, with the inverse element relative to multiplication for all elements except zero.

In a ring and a field, the properties of operations are often similar to addition and multiplication in the set of real numbers or the set of computer arithmetic. However, for a number of algebras, it is more convenient to use the operations of union ($\cup$) and intersection ($\cap$).

**Definition 5.** A lattice is an algebra $\langle X; \cup, \cap \rangle$ that has a number of properties:

1. Idempotency—$x \cap x = x, x \cup x = x$;
2. Absorption—$(x \cap y) \cup x = x, (x \cup y) \cap x = x$;
3. Absorption—$(x \cup y) \cup x = x$
4. Associativity with respect to both operations;
5. Commutativity with respect to both operations;
6. Partial order defined on the lattice.

If the distributivity property is satisfied, the lattice is also called distributive.

Now, let us proceed directly to the definition of the algebraic structure for a set of signatures of the same digit.

3. Results
3.1. Operations on First-Order Signatures

When solving the problem of determining the operations of merging and intersection of signatures, we must realize that the efficiency of standard algorithms based on adjacency matrices is rather inefficient, because the processing of signatures is linear with respect to the number of vertices in the graph, and algorithms based on adjacency matrices have a quadratic dependence on the number of vertices. Therefore, when defining the algebra of signatures, it is desirable to define operations on them as isolated from other signature-related objects such as graphs, adjacency matrices, vertex degree vectors, and bases. Accordingly, for the definitions of union and intersection of signatures, let us construct a union algorithm.

Let us proceed to the description of the algorithms for the union and intersection of signatures.
The essence of the algorithm for combining two signatures is as follows: first, we compare the decimal values of the signatures, and define the larger signature as the first one. Then, we go bit by bit through the signatures to be merged, starting from the highest digit and moving to the lowest. Each pair of digits is compared. If they are equal, the same value is set in the result bit as in the compared bits. If the digits are different, the accumulated difference between the digits is taken into account. The resulting digit is set to the same value as the same signature digit with the prevailing difference. After that, the difference between the current digits of the first and second signatures is added to the difference.

Let us consider the action of Algorithm 1 in an example.

Algorithm 1 Union operation for two signatures

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Set ( d = 0 ), ( i = 0 ).</td>
</tr>
<tr>
<td>2.</td>
<td>If ( \text{sign1} &lt; \text{sign2} ), we swap them: ( \text{sign1}, \text{sign2} = \text{sign2}, \text{sign1} ).</td>
</tr>
<tr>
<td>3.</td>
<td>If the ( i )-th bit of ( \text{sign1} ) is greater than or equal to the ( i )-th bit of ( \text{sign2} ), then write the value of ( \text{sign1}[i] ) into the ( \text{sign3} ) the value of ( \text{sign1}[i] ), otherwise go to Step 5.</td>
</tr>
<tr>
<td>4.</td>
<td>If the ( i )-th bit of ( \text{sign1} ) is greater than the ( i )-th bit of ( \text{sign2} ), then increase ( d ) by 1 and ( i ) by 1 and go to Step 3, otherwise increase ( i ) by 1 and go to Step 3.</td>
</tr>
<tr>
<td>5.</td>
<td>Decrease ( d ) by 1.</td>
</tr>
<tr>
<td>6.</td>
<td>If ( d &lt; 0 ), then swap the signatures: ( \text{sign1}, \text{sign2} = \text{sign2}, \text{sign1} ), multiply ( d ) by (-1).</td>
</tr>
<tr>
<td>7.</td>
<td>Write the value of ( \text{sign1}[i] ) into ( \text{sign3} ), increase ( i ) by 1 and go to Step 3.</td>
</tr>
<tr>
<td>8.</td>
<td>The signature of the union of two signatures has been obtained.</td>
</tr>
</tbody>
</table>

Example 3. Let \( \sigma_1 = 101110_2 \) (see Figure 6) and \( \sigma_2 = 110101_2 \) (see Figure 7) be given as input. First, let us check whether the input data meet the condition \( \sigma_1 < \sigma_2 = 1011102 < 1101012 \); then, we swap \( \sigma_1, \sigma_2 \), respectively, and obtain \( \sigma_1 = 1101012 \) and \( \sigma_2 = 1011102 \). Now, we can go step-by-step through the algorithm and follow the conditions. Let us denote that the zero bit will be the very first bit of the number. The further process of finding \( \sigma_3 \) is summarized in Table 4 with the following lines: \( i, d \)—initial value before the loop; \( \sigma_1, \sigma_2, \sigma_3 \) and \( d^* \)—current value during the loop. The union graph is shown in Figure 8.

Figure 6. Graph described by signature \( \sigma_1 = 101110_2 \) from Example 3.
Next, let us consider the Algorithm 2 for the intersection of signatures, which is very similar to the union algorithm, but differs in some actions.

The essence of the algorithm for the intersection of two signatures is as follows: first, we compare the decimal values of the signatures, and define the smaller signature as the first one. Then, we go bit-by-bit through the intersected signatures, starting from the higher digit and moving to the lower one. Each pair of digits is compared. If they are equal, the same value is set in the result bit as in the compared bits. If the digits are different, the accumulated difference between the digits is taken into account. The resulting digit
is set to the same value as the same signature digit with a smaller difference. After that, the difference between the current digits of the first and second signatures is added to the difference.

**Algorithm 2** Crossing operation for two signatures

```
Input data: sign1, sign2-signatures (non-negative integer).
Local variables: index 0 <= i <= length of sign1, d - counting the number of units (at 0 1 and 1 0).
Output: sign3-signature (non-negative integer).
Step 1. Set d = 0, i = 0.
Step 2. If sign1 > sign2, then swap them: sign1, sign2 = sign2, sign1.
Step 3. If the i-th bit of sign1 is less than or equal to the i-th bit of sign2, then write the value of sign1[i] to sign3, otherwise go to Step 5.
Step 4. If the i-th bit of sign1 is less than the i-th bit of sign2, then increase d by 1 and i by 1 and go to Step 3, otherwise increase i by 1 and go to Step 3.
Step 5. decrease d by 1.
Step 6. If d < 0, then swap the signatures: sign1, sign2 = sign2, sign1, multiply d by −1.
Step 7. Write the value of sign1[i] into sign3, increase i by 1 and go to Step 3.
Step 8. The intersection of the two signatures has been obtained.
```

Let us consider the action of the algorithm on an example.

**Example 4.** Let sign1 = 111011 (see Figure 9) and sign2 = 101010 (see Figure 10) be given as input. First, let us check whether the input data meet the condition sign1 > sign2 = 111011 > 101010; then, we swap sign1 and sign2, respectively, and obtain σ1 = 101010 and σ2 = 111011. Now, we can go step-by-step through the algorithm and follow the conditions. Let us denote that the zero bit will be the very first bit of the number. The further process of finding sign3 is shown in Table 5 with the following lines: i, d—the initial value before the loop; σ1, σ2, σ3, and d*—the current value during the loop.

In this example, the resulting signature is the same as the σ2 signature.

![Figure 9](image-url)  
*Figure 9. Graph described by signature sign2 = 111011 from Example 4.*
As we can see from the algorithms and examples, the larger number for sign1 is chosen for union, and the smaller number for intersection is chosen for intersection. And when the number of consecutive units in any of the signatures increases, we swap them. Working with these algorithms simplifies the work with graphs, because now not only can we store them compactly to make more memory available, but also work on operations compactly without going beyond the signatures.

3.2. Algebraic Structures on the Set of Signatures

Let us start defining structures from the least constrained to the most constrained.

**Theorem 1.** Let $\Sigma_n$ be the set of all signatures with $n$ digits. Then, the algebra $(\Sigma_n \cup)$ is a semigroup with respect to the union operation.

**Proof.** Let the signatures $\sigma_x$, $\sigma_y$, and $\sigma_z$ correspond to the matrices $X$, $Y$, and $Z$. According to Definition 1, the adjacency matrix $V = (v_{ij})$ corresponding to the signature $\sigma_v = \sigma_x \cup \sigma_y$ can be constructed by the following rule:

$$v_{ij} = \max(x_{ij}, y_{ij}).$$

(1)

According to the associativity rule, the following expression must be true for any signature:

$$(\sigma_x \cup \sigma_y) \cup \sigma_z = \sigma_x \cup (\sigma_y \cup \sigma_z).$$

(2)

Since the signature corresponds to its adjacency matrix, (2) can be represented by the following equality:

$$\max(\max(x_{ij}, y_{ij}), z_{ij}) = \max(x_{ij}, \max(y_{ij}, z_{ij})).$$

(3)
At the same time, it is known that the operation of taking the maximum is associative, which allows us to transform (3) into
\[
\max(\max(x_{ij}, y_{ij}), z_{ij}) = \max(x_{ij}, y_{ij}, z_{ij}) = \max(x_{ij}, \max(y_{ij}, z_{ij})).
\] (4)
Thus, the matrix obtained from the elements of the left part of the Equation (4) is equal to the matrix obtained from the elements of the right part of the same equation. Moreover, these matrices correspond to the signatures obtained in the left and right parts of the (2) equation, respectively. Consequently, Equation (2) is an identity, and the associativity rule of the operation of union signatures is fulfilled.

The following signature intersection theorem is defined similarly.

**Theorem 2.** Let \( \Sigma_n \) be the set of all signatures with \( n \) digits. Then, the algebra \( \langle \Sigma; \cap \rangle \) is a semigroup with respect to the intersection operation.

**Proof.** By analogy with the proof of Theorem 1, let the signatures \( \sigma_x, \sigma_y, \) and \( \sigma_z \) correspond to the matrices \( X, Y, \) and \( Z. \) According to Definition 1, the adjacency matrix \( W = (w_{ij}) \) corresponding to the signature \( \sigma_v = \sigma_x \cup \sigma_y \) can be constructed by the following rule:
\[
W_{ij} = \min(x_{ij}, y_{ij}).
\]

According to the associativity rule, the following expression must be true for any signature:
\[
(\sigma_x \cap \sigma_y) \cap \sigma_z = \sigma_x \cap (\sigma_y \cap \sigma_z),
\]
Further reasoning is fully equivalent to the proof of Theorem 1, and proves the associativity of the signature intersection operation.

To specify the following structure, we can define two signatures: \( \sigma_0 = 0 \) and \( \sigma_U = 2^n - 1, \) with all digits equal to 1.

**Theorem 3.** Let \( \Sigma_n \) be the set of all signatures with \( n \) digits. Then, the algebra \( \langle \Sigma; \cup \rangle \) is a monoid with respect to the union operation with \( \sigma_0 \) as a identity element.

**Proof.** Performing the operation on the signatures \( \sigma \) and \( \sigma_0, \) we will see that if there is some number of first bits in \( \sigma \) and \( \sigma_0, \) they will be equal. If the first signature is not equal to 0, one of its digits will be equal to 1. Starting from this digit, the difference between the digits of the signatures will always be positive, which causes the same values for all digits in the unity signature as in the first signature.

**Theorem 4.** Let \( \Sigma_n \) be the set of all signatures with \( n \) digits. Then, the algebra \( \langle \Sigma; \cap \rangle \) is a monoid with respect to the intersection operation with \( \sigma_U \) as a identity element.

Similarly, when dealing with the intersection operation, we select the full signature for the identity element.

**Proof.** Performing the operation on the signatures \( \sigma \) and \( \sigma_U, \) we will see that if there are some number of first going single digits \( \sigma \) and \( \sigma_0 \) will be equal. If the first signature is not equal to \( \sigma_U, \) one of its digits will be equal to 0. Starting from this digit, the difference between the digits of the signatures will always be negative, which causes the same values in the intersection signature for all digits as in the first signature.

Now, consider the next construct: a group.
The definition of a monoid requires the existence of an inverse element to an identity element. Determining the inverse element to an arbitrary signature in such a way that their union leads to a null signature is not yet feasible, which leads to the following theorem.

**Theorem 5.**

1. Monoid \( \langle \Sigma; \cup \rangle \) with identity element equal to \( \sigma_0 \) does not form a group.
2. Monoid \( \langle \Sigma; \cap \rangle \) with identity element equal to \( \sigma_u \) does not form a group

**Proof.** A group requires the existence of inverse elements for each signature. The operation of combining a signature with its inverse must yield an identity element, which we defined earlier as \( \sigma_0 \). In this case, \( \sigma_0 \) consists only of zero bits, which means that the following union of two signatures will lead to \( \sigma_0 \) only if they themselves are equal to \( \sigma_0 \). We can come to similar conclusions by analyzing the resulting monoid on the set of signatures with the operation of permutation. ✷

Obviously, it is meaningless to consider a ring and a field with a constructed algebra based on the union and intersection operations, since they require a group on the operations in question.

### 3.3. Distributive Lattice

Recall that the set of extremal graphs with union and intersection operations form a distributive lattice. Since the set of signatures is a representation of the set of extremal graphs, it is logical to assume that the set of signatures also forms a distributive lattice. For this purpose, let us introduce a theorem and prove that all properties of the distributive lattice are satisfied.

To define a distributive lattice, we need to define a partial order on the signature sets and make sure that all properties of the distributive lattice are satisfied.

Let us first define the partial order. To construct the partial order, let us look at the logic of signature construction once again.

The highest zero digit means that the vertex with the highest number is at least isolated in the graph. If we remove this vertex and correspondingly remove one last row and column from the adjacency matrix, we obtain the same signature, but one digit less. Similarly, we can increase the number of vertices in the graph by adding any number of isolated vertices. The signature will not change numerically; only the number of available digits that will be added will change, and they will remain zero.

We will assume that the signatures \( \sigma_1 \) and \( \sigma_2 \) describing the adjacency matrices \( X_1 \) and \( X_2 \) are of partial order \( \sigma_1 \leq \sigma_2 \) if and only if \( \forall (p,q) \in \{(i,j) | x_{1ij} = 1\} \exists x_{2pq} = 1 \) and \( \not\exists x_{2pq} = 0 \). Signatures are equal only when the adjacency matrices they describe are equal.

After analyzing the signature properties, we can introduce the partial order relation without referring to the adjacency matrices.

Let us denote by \( \sigma(k) \) the \( k \)-th digit of the signature.

**Definition 6.** Let two signatures \( \sigma_1 \) and \( \sigma_2 \) be given, on \( n \) digits each. We will assume that \( \sigma_1 \leq \sigma_2 \) if for \( \forall p \geq 0 \sum_{p}^{n-1} \sigma_1(p) \leq \sum_{p}^{n-1} \sigma_2(p) \).

**Theorem 6.** Let \( \Sigma_n \) be the set of all signatures with \( n \) digits. Then, the algebra \( \langle \Sigma_n; \cup, \cap \rangle \) is a distributive lattice with respect to intersection and union operations.

**Proof.**

1. Idempotency: By definition of union and intersection operations, if all digits in the original signatures are equal in pairs, then the resulting signature is also equal to any of the signatures. That is, the application of union and intersection operations on identical signatures does not lead to a new signature, but remains the same as the old.
2. Associativity: As shown in Theorems 1 and 2, the associativity property for the signature algebra holds.
3. Commutativity: This property comes from the definition of union and intersection operations, where before the direct finding of the result of the operation, the signatures were first ordered according to the value, which leads to the same process of calculating the union or intersection of the signatures.

4. Absorption: Let us describe the absorption property through operations on adjacency matrices.

\[(x \cap y) \cup x = x\]
\[
\max_{ij} (\min_{ij} (x_{ij}, y_{ij}), x_{ij}) = x_{ij}.
\]

According to the constructed expression, we can easily prove the enforceability of the absorption property. If \(x_{ij} < y_{ij}\), then \(\min\) will give us the values of the matrix \(X\). The same reasoning is true for the following formula:

\[(x \cap y) \cup x = x.
\]

5. Distributivity of the lattice: let us also represent the distributivity property through operations on the adjacency matrices:

\[x \cup (y \cap z) = (x \cup y) \cap (x \cup z),\]
\[
\max_{ij} (\min_{ij} (y_{ij}, z_{ij})) = \min_{ij} (\max_{ij} (x_{ij}, y_{ij}), \max_{ij} (x_{ij}, z_{ij})).
\]

Thus, it is shown that the signature algebra forms a distributive lattice.

Due to the establishment of partial order on the signature set, the exact and lower bounds can be determined.

\[
\inf \Sigma_n = \sigma_0 = 0 \ldots 0_{\frac{n}{n}}
\]
\[
\sup \Sigma_n = \sigma_{\Omega} = 1 \ldots 1_{\frac{n}{n}}.
\]

Identification of the lower and upper edges, as well as the description of the principle of their formation, will allow us to further describe the mechanism of construction of the Hass diagram for a set of signatures.

4. Conclusions

In this paper, we consider algebras on signatures with union and intersection operations. The latter two operations were first defined only through signature and bitwise operations. However, the other class of operations—arithmetic operations—was not considered here, which is quite possible. However, the meaning of using signature addition and multiplication operations is not clear yet. Actually, the main doubt of the authors is that the practical application of these operations has not yet been found.

It should also be noted that we assert the impossibility, in this configuration, to construct a group, and, accordingly, a ring and a field on the set of signatures. This assertion relies on an undefined process of finding an inverse element to a signature.

It seems to the authors that this problem can be circumvented either by extending the set of signatures or by introducing operations modulo \(2^n - 1\) on the space of signatures, where \(n\) is the power of the set of vertices of an extremal two-uniform hypergraph corresponding to the signature under consideration.

Further work in this area will undoubtedly extend the methods for analyzing and interacting with signatures.

The practical application of this study can be found in various areas of mathematics, for example, in the case of extending the obtained results to normalized [26] and Laplassian [27] matrices.
The same practical application is possible in new topological encryption methods [10] where the key is a hypergraph or its characteristics, such as signatures. Transformations over extremal two-uniform hypergraphs in general take quadratic time, and the transition to operations on signatures will reduce the problems of union or intersection of key hypergraphs to linear complexity.

It is also of great interest to construct algebras on the sets of extremal vectors of degrees of vertices, the set of bases and the set of second-order signatures. It is supposed that this will allow one to make generalizations about the internal structure of the class of extreme $k$-uniform hypergraphs.

**Author Contributions:** Conceptualization, A.M.; validation, E.E., V.T.; formal analysis, E.E.; investigation, A.M.; writing—original draft preparation, A.M.; writing—review and editing, V.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are contained within the article.

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

**Abbreviations**
The following abbreviations are used in this manuscript:

- **UH** Uniform hypergraph
- **EUH** Extreme uniform hypergraph

**References**


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