Thai Journal of **Math**ematics
Volume 21 Number 4 (2023)
Pages 1061–1079

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Discrete and Computational Geometry, Graphs, and Games

Clarifying the Difference between Origami Fold Models by a Matrix Representation

Yiyang Jia[1,*], **Jun Mitani**[2] **and Ryuhei Uehara**[3]

[1] *Seikei University, Japan*
e-mail : jiayiyang@st.seikei.ac.jp

[2] *University of Tsukuba, Japan*
e-mail : mitani@cs.tsukuba.ac.jp

[3] *School of Information Science, JAIST, Japan*
e-mail : uehara@jaist.ac.jp

Abstract In this paper, we investigate the accessible flat-folded states of three common fold models under the context of two origami problems. The three fold models are the simple fold model, the simple fold-unfold model, and the general fold model. The two problems are the valid total order problem (VTP) and the valid boundary order problem (VBP). They both belong to the field of computational origami and are considered as two variants of the map folding problem. As a result, in both VTP and VBP, the proper inclusion relationship between the sets of the accessible valid orders of the simple fold model and the simple fold-unfold model, and the proper inclusion relationship between the sets of the accessible valid orders of the simple fold-unfold model and the general fold model are clarified. We first introduce a four-dimensional logical matrix representation to prove these proper inclusions mathematically. Then, we further explain the proper inclusion relationship by actual map folding examples. This work extends our previous result: using the logical matrix representation to indicate arbitrary map foldings, where we have discussed the logical matrix representation from the viewpoint of category theory. This time, instead of theoretical analysis, we realign the logical matrix to a four-dimensional form and use it to investigate VTP and VBP.

MSC: 49K35; 47H10; 20M12
Keywords: fold models; computational origami; valid order problem; logical matrix representation

Submission date: 30.01.2022 / Acceptance date: 23.02.2023

1. Introduction

Flat folding is one of the most critical topics in computational origami. These foldings are categorized with respect to their final flat-folded states, which are supposed to be embedded in a two-dimensional plane. Existing relevant researches can be roughly clustered

*Corresponding author.

Published by The Mathematical Association of Thailand.
Copyright © 2023 by TJM. All rights reserved.

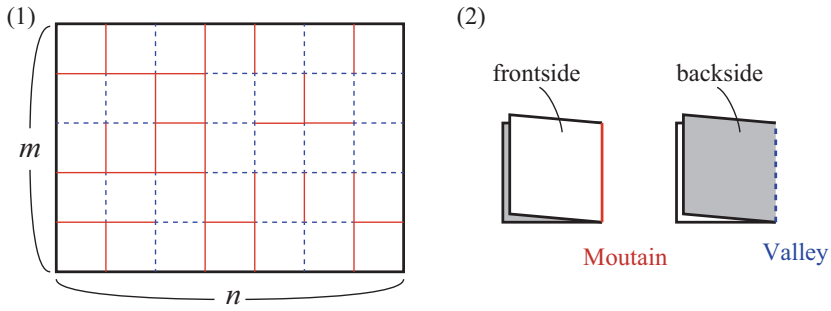

FIGURE 1. A map instance of the map folding problem. The red solid-line segments represent mountains; and the blue dashed-line segments represent valleys.

into two types. One investigates whether a known *crease pattern* (a planar graph embedded in a piece of paper that is supposed to be folded along the edges of the graph) with or without the *Mountain-Valley assignment* (an assignment which maps each edge of the crease pattern to either a *mountain* or a *valley*, representing the two folding directions as illustrated in Figure 1 (2), respectively) can be folded to a flat state. The other concerns how many possible flat-folded states exist. The first problem type was addressed as a decision problem and proved NP-hard in general cases[1]. However, in some special cases, for example, when the crease pattern is simplified to a grid pattern whose every face is a unit square and with an MV-assignment, there is still no proof of whether this case also belongs to NP-hard or not. This problem is called the *map folding problem*, proposed by Jack Edmonds in 1997 in a personal talk. Precisely, it asks the time complexity to decide whether a given grid pattern with all its edges assigned the folding directions (mountains and valleys) can be folded to the shape of a unit square or not. In Figure 1(1), we give an instance of this problem.

Map folding is considered a fundamental branch in the field of flat folding. Even though the map folding problem remains unsolved now, many of its variants have been investigated and worked out. Some results can be found in, e.g., [2–4]. Among the variations, two important decision problems, the *valid total order problem (VTP)* and the *valid boundary order problem (VBP)*, will be discussed in this paper. VTP asks if a given order of all the squares of the map corresponds to a *valid overlapping*, i.e., the order of the squares in an accessible flat-folded state from bottom to top, as illustrated in Figure 2. The other problem, VBP, asks whether a given order of the squares on the boundary of the map corresponds to some flat-folded state or not. An instance is illustrated in Figure 3.

At the same time, three different fold models, including the *simple fold model*, the *simple fold-unfold model*, and the *general fold model*, can be applied to both VTP and VBP. The definitions of these models will be introduced in Section 2. Some known results about these different fold models can be found in [3, 5]. However, the space of accessible overlapping orders, i.e., the set of orders of squares corresponding to accessible flat-folded states of different fold models, still lacks a mathematical analysis.

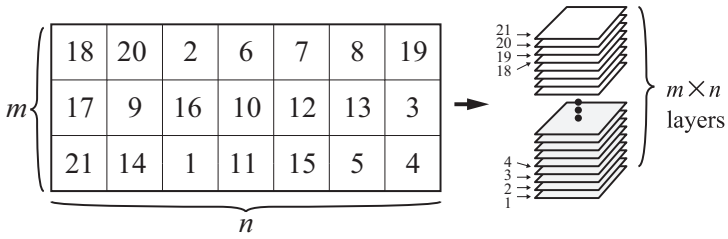

FIGURE 2. An instance of VTP: a map is folded into $m \times n$ square layers. The overlapping order of the squares in the folded state is given.

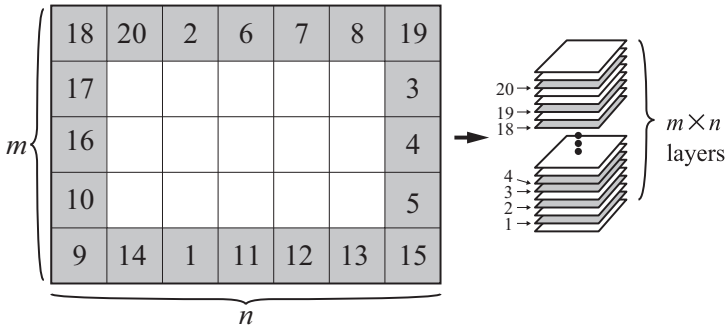

FIGURE 3. An instance of VBP: a map is folded into $m \times n$ square layers. The overlapping order of the boundary squares in the folded state is given.

In this paper, we investigate the sets of accessible overlapping orders of three different fold models and their relations. Our result is illustrated in Table 1. This result has been once introduced as a by-product of the solution of these combined problems in [6]. However, a clear mathematical analysis or proof was never given yet. We classify the proper inclusion relationships (as shown in Table 1) by introducing a four-dimensional logical matrix representation of the map folding. In previous research, we have used two-dimensional logical matrix representations to indicate map foldings. The idea of using logical matrix representations originates from the viewpoint of category theory. Such matrix representations offer us a way to associate map folding with the category of relations, **Rel**, which further allow us to apply common algebraic tools (such as the constructions of linear structures, semi-modules, and tensors) to study map folding [12, 13]. Moreover, the extension of **Rel** to other *monoidal categories* inspired us to extend the representations and solutions of map folding to more general flat foldings [7]. In this paper, we apply the logical matrix representation to analyze the fold models and variations of map foldings from the viewpoint of computational origami. We adjust the matrix model to a four-dimensional one such that the folding process of different fold models can be clearly traced by these matrices. In conclusion, we managed to prove the proper inclusion relationships shown in Table 1. As an assurance, we also propose some instances in the relative complement sets to verify our matrix analysis.

Problems	Simple Fold		Simple Fold-Unfold		General Fold
VTP	$\boldsymbol{A_1}$	$\subsetneq$	$\boldsymbol{A_2}$	$\subsetneq$	$\boldsymbol{A_3}$
VBP	$\boldsymbol{B_1}$	$\subsetneq$	$\boldsymbol{B_2}$	$\subsetneq$	$\boldsymbol{B_3}$

TABLE 1. The results of three fold models in VTP and VBP.

In the following, we will provide the preliminaries of this work in Section 2. In Section 3, we will first introduce the two-dimensional logical matrix representation of $1 \times n$ maps and then adjust such a logical matrix representation to four-dimensional to represent $m \times n$ maps. Next, we will use the four-dimensional logical matrix representations to prove the proper inclusion relationships illustrated in Table 1. In Section 4, we will propose practical VTP and VBP instances to verify our conclusion, respectively. Finally, we will conclude this work and describe possible future work in Section 5. Our research reflects the importance of logical matrix representation. Essentially, it associates flat foldings with category theory, tensors, and other algebras in a theoretical way. Moreover, the logical matrix representation can be directly applied to analyze problems in computational origami[8], where the problems were traditionally expressed by graphs and then solved by additional computations.

2. PRELIMINARIES

The definitions and notations used in this paper and the known results of relevant problems are introduced in this section.

2.1. DEFINITIONS AND NOTATIONS

Definition 2.1. A *crease pattern* C is a straight-line planar graph (V_C, E_C) together with the compact connected closed subset of $\mathbb{R}^2$ where (V_C, E_C) is embedded in. The closed subset represents a piece of paper. Every $e \in E_C$ is called an *edge*. Every maximal region bounded by either the boundary of the paper or some edges in E_C without any crease inside is called a *face*. A *Mountain-Valley assignment (an MV assignment)* is a function $f : E_C \to \{M, V\}$, where every edge is mapped to a folding direction, as shown on the right side of Figure 1.

A *flat folding* or a *flat-folded state* consists of an isometric geometry function mapping each point of paper to a point in 2D and an ordering function specifying for each pair of non-crease points mapping to the same 2D point, which point is on top while avoiding crossings [8]. A *final state* refers to a flat-folded state corresponding to a given crease pattern with or without the MV assignment, where all the creases in the crease pattern are supposed to be folded (following the assigned directions when the MV assignment is also given). A *middle state* refers to a partly folded state of a crease pattern, i.e., a flat-folded state with only part of the creases in the given crease pattern folded. If either a final state or a middle state of a map can be obtained using a certain fold model, we say that such a state is *accessible* by this model.

In our illustrations, an $m \times n$ map is represented by a rectangle sheet with m rows and n columns made up of $m \times n$ congruent squares. The map is simplified to one dimensional when m equals one and will be indicated by a line segment with squares as segments of unit length. We also have the following definitions.

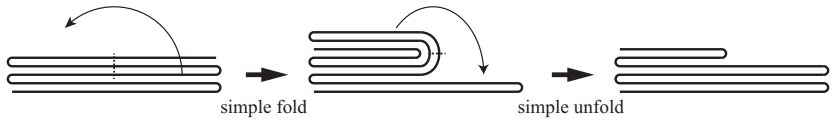

FIGURE 4. An example of simple fold and unfold(cross-section).

Definition 2.2. The edges of the squares in a map not located at the boundary of the map together comprise a *grid pattern* as the crease pattern of the map. *Map folding* is a specific type of flat folding by restricting the crease pattern C to a grid pattern with a given MV assignment (as illustrated in Figure 1). In the map folding, all the faces are unit squares. Two squares sharing the same crease are called a pair of *neighbor* squares.

Definition 2.3. In any middle state or the final state, we say a pair of squares whose surfaces touch each other are *adjacent*. We use *overlapping orders* to refer to the orders given on the set of all the squares of the map (partial and total orders). If such an input order corresponds to the order of these squares aligning from bottom to top in an accessible final state of the map with respect to a certain fold model, then it is said to be *valid*, and if such an input order corresponds to an inaccessible folded state of the map, i.e., a state either with some self-penetrations or that cannot be reached by the certain fold model, it is called *invalid*.

The following map folding problem was originally given by Jack Edmonds in a personal talk in 1997 [8].

Definition 2.4. (*Map Folding Problem*) Is it possible to flatly fold a map of size $m \times n$ with a specific MV-assignment? Furthermore, what is the computational complexity of determining whether it has a flat folding or not?

The following three definitions are about different fold models.

Definition 2.5. A *general fold model* is a fold model without additional constraints as long as the paper is not stretched, torn, or penetrated.

Definition 2.6. A *simple fold* is a folding operation that folds contiguous layers along a single line segment to a certain direction (either all mountain folds or all valley folds) and does not change the flat state of other layers. In other words, only the properties in arbitrarily small neighborhoods of points on the folded segment would be changed. A *simple fold model* is a fold model that permits only simple folds.

Definition 2.7. A *simple fold-unfold model* is a fold model that permits both simple folds and the reverse operation of simple folds, *simple unfolds*.

Examples of a simple fold and a simple unfold are illustrated in Figure 4.

Now, we introduce the two main problems dealt with in this paper, **VTP** and **VBP**.

Definition 2.8. *Valid Total Order Problem (VTP)*: Given an order of all the squares of a map, is there an accessible final state of the map where these squares align from bottom to top in this order?

Definition 2.9. *Valid Boundary Order Problem (VBP)*: Given an order of only the squares lying on the boundary of a map, is there an accessible final state of the map where these squares align from bottom to top in this order?

Other notations will be given in respective sections.

2.2. Relevant Results

In general, determining whether a crease pattern (with or without an MV assignment) has a flat folding or not is proved to be NP-hard [1]. Then, even simplified a lot, determining whether a grid pattern with arbitrarily existing diagonals (with or without a certain MV-assignment) has a flat folding or not has also been proved NP-hard [2]. Finally, the map folding problem whose object is a grid pattern with a certain MV-assignment still remains unsolved.

Since [8] verified that a crease pattern is flat-foldable if and only if it has a flat final state (regardless of the folding process), some researchers have investigated the possible final states rather than the crease patterns. In [3], they have proposed a linear time solution to VTP with the general fold model. Their solution is finished by checking all the penetrating pairs of squares in the state represented by the given order. This result implies that the map folding problem must belong to NP. As afore-mentioned, this problem can also be combined with the settings of other fold models. [9] proposed a linear-time algorithm ($\mathrm{O}(mn)$) to solve VTP when employing the *simple fold model*. The simple fold model increases the difficulty of VTP because not only the folded states but also the whole folding process must be considered. It must be guaranteed that every fold during the folding process is a simple fold. The third model, the simple fold-unfold model, was firstly defined in [4]. By replacing the step that verifies one-dimensional simple folds in the algorithm presented in [9] with the one-dimensional simple fold-unfold verification in [4], VTP with the simple fold-unfold model can also be solved in linear time. In this paper, we focus on mathematically proving the proper inclusion relationship between the sets of accessible flat-folded states of each two models, as shown in the first row of Table 1.

The other problem, VBP, was firstly presented in [5] (Figure 3). This problem is also solvable in time linear to the input ($\mathrm{O}(m+n)$) when applying the simple fold model [5]. With some adjustments, the algorithm provided in [5] can also deal with VBP using the simple fold-unfold model and the general fold model in time linear to the size of the input ($\mathrm{O}(mn)$). As in VTP, the proper inclusion relationship between the sets of accessible flat-folded states of each two models, as shown on the second row of Table 1, is the crucial point in this research.

We will use a four-dimensional logical matrix representation of the map folding to clarify the inclusion relationship in both VTP and VBP with these three fold models. In some of our other research, we presented the logical matrix representation to associate the flat-folded states of a map with the category of relations (**Rel**). This representation further shows the possibility to associate the flat-folded states of general crease patterns with the category of finite Hilbert spaces (**FHilb**), which is a category principally connected with the field of quantum computing [12]. Researches have shown that (bi)linear structures can be built in monoidal categories. If we combine the linear algebraic structures that we can construct on the logical matrix (e.g., matrix semiring, tensor built from the monoidal category), then based on the ideas in the famous Grothendieck's Tohoku paper [14], we are hopeful to establish a new homology or cohomology on this structure just like the sheaf cohomology. Such mathematical analysis would then give rise to a clarification on the connection between the local and global properties [15]. Despite that the analysis of flat folding in category theory is inspiring, we still expect the usefulness of logical matrix

representation when applying it to computational origami. In this research, we will show that the logical matrix representation can be employed to explain the inclusion relationship of the sets of accessible flat final states by different fold models in VTP and VBP clearly.

3. Logical Matrix Representation of $1 \times n$ and $m \times n$ Maps

In this section, we will first introduce the logical matrix as a representation of relations. Next, we will propose a two-dimensional logical matrix representation of $1 \times n$ maps. Instead of directly expressing the folding motion, this representation expresses it by recording every middle or final state of the map after a phase during the folding. The adjacency between squares can represent a middle state, i.e., two squares are either adjacent or not adjacent, which matches the 0s and 1s in the corresponding logical matrix. Then, we will extend the two-dimensional matrices to four-dimensional ones to describe the foldings of $m \times n$ maps, which provides the basic structure to analyze the accessible final states of different models by comparing their corresponding matrix spaces.

3.1. Relations and Logical Matrices

Logical matrices naturally have the algebraic structure as matrix semirings. They are built over the *boolean semiring* $\Omega = (\{0, 1\}, +, \cdot)$. When this semiring is associated with the relations, its elements 1 and 0 correspond to logical true (relation exists) and logical false (relation does not exist), respectively. The ring operations of the logical matrices are inherited from Ω and are applied to the matrix elements independently. The addition $+$ and the multiplication $\cdot$ on Ω are defined as follows.

$$\begin{array}{ll} \Omega \times \Omega \to \Omega & \Omega \times \Omega \to \Omega \\ (0,0) \mapsto 0+0=0 & (0,0) \mapsto 0 \cdot 0 = 0 \\ (0,1) \mapsto 0+1=1 & (0,1) \mapsto 0 \cdot 1 = 0 \\ (1,0) \mapsto 1+0=1 & (1,0) \mapsto 1 \cdot 0 = 0 \\ (1,1) \mapsto 1+1=1 & (1,1) \mapsto 1 \cdot 1 = 1 \end{array} \tag{3.1}$$

The next definition introduces the logical matrix representation of relations.

Definition 3.1. Let $R(A, B)$ be a relation over A and B with $A = \{a_1, a_2, \cdots, a_n\}$ and $B = \{b_1, b_2, \cdots, b_m\}$. Then an m-row n-column matrix M is a logical matrix representation of $R(A, B)$ if and only if $M_{ij} = 1$ when $(a_j, b_i) \in R(A, B)$ and $M_{ij} = 0$ when $(a_j, b_i) \notin R(A, B)$.

Example 3.2. For a relation R=$\{a_1 R b_2,\ a_3 R b_2,\ a_3 R b_4\}$ between the sets A=$\{a_1,\ a_2,\ a_3\}$, B=$\{b_1,\ b_2,\ b_3,\ b_4\}$, its corresponding logical matrix is defined as below.

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The relationship between a relation and its logical matrix representation is described as follows.

Lemma 3.3. *(1) For two relations $R_1, R_2 \subseteq A_0 \times A_1$ and their corresponding logical matrices M_1 and M_2, resp., the relation $R = R_1 \cup R_2$ corresponds to the logical matrix $M = M_1 + M_2$.*
(2) For relations $R_1 \subseteq A_0 \times A_1$ and $R_2 \subseteq A_1 \times A_2$, if their corresponding logical matrices are M_1 and M_2, resp., then their composition relation $R \subseteq A_0 \times A_2$ obtained as

$$\{(a_0, a_2) | \exists a_1 \in A_1 \text{ s.t., } (a_0, a_1) \in R_1 \text{ and } (a_1, a_2) \in R_2\}$$

corresponds to the logical matrix $M = M_2 \cdot M_1$.
The addition and multiplication follow the operations over Ω.

Proof. The above lemma is straightforward according to that the boolean value 1 indicates the existence of the corresponding relation. ∎

The following paragraphs introduce the *transitive closure* of relations and how we can obtain them by matrix computations.

Definition 3.4. The **transitive closure** of a binary relation $R \subseteq A \times A$ is the smallest relation on A that contains R and is transitive. In this expression, "transitive" means that for any $(a_0, a_1), (a_1, a_2) \in A$, $(a_0, a_2) \in A$ holds.

Let sets A_0, A_1, A_2 in Lemma 3.3 refer to the same set, we can further obtain the following theorem as a generalized conclusion.

Theorem 3.5. *The transitive closure R^+ of R corresponds to the addition of powers of its logical matrix. That is, if the logical matrix representation of R is an $n \times n$ matrix M, then the logical matrix representation of R^+ is $\sum_{i=1}^{n-1} M^i$.*

Proof. The correspondence between the transitive closure of a relation and the power sum of its logical matrix follows the introduction of the meaning of the transitive closure in [10]. Also, by computation, for any $n' > n-1$, we have $\sum_{i=1}^{n'} M^i = R^+ = \sum_{i=1}^{n-1} M^i$. ∎

3.2. Logical Matrix Representation of $1 \times n$ maps

The basic idea of constructing a logical matrix representation is taking the "up-down adjacency" between every pair of squares as the relation. In this way, any "up-down relation" between every pair of squares can be obtained from the transitive closure. We will present the logical matrix corresponding to the "up-down adjacency", the logical matrix corresponding to the "up-down relation", and the logical matrix representation of arbitrary middle states of $1 \times n$ maps in order.

In the following two paragraphs, we first give the logical matrix representations of possible final states of the map. Then, we will further show how to use logical matrix representations to represent middle states, i.e., where a map is only partly folded. The last one will be applied to the analysis of the simple fold model and the simple fold-unfold model.

Representation of the up-down adjacency. For any final state of a $1 \times n$ map, we can record the "up-down adjacency" between every pair of squares in a logical matrix. Firstly, for any final state of a $1 \times n$ map with the i-th square folded adjacently below the j-th square, we record $i < j$ as the relation for this pair. Then, we can obtain the corresponding logical matrix of all such relations. The assigning of 0s and 1s follows the method given in Definition 3.1. For example, as illustrated in Figure 5 (b) and (c),

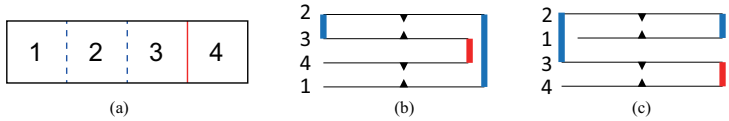

FIGURE 5. Two possible final states of a 1×4 map. The blue line segments and the red line segments indicate the creases. The triangles indicate the frontside of the map.

two possible final states of the map in (a) correspond to P_1: $1 < 4 < 3 < 2$, and P_2: $4 < 3 < 1 < 2$, respectively. Their corresponding logical matrices are then built as follows.

For P_1 and P_2, we have

$$P_1 : \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad P_2 : \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

An interesting observation is that an all-elements 0 assignment of the i-th row means that the square indexed i will be at the bottom of all the squares. Similarly, an all-elements 0 assignment of the j-th column means that the square indexed j will be at the top of all the squares.

Representation of the up-down relation. The above logical matrix representation of the up-down adjacency is, however, not enough in the sense of globally capturing the properties of a folded state. We thus further build a representation of the up-down relation. As introduced before, an up-down relation is just a transitive closure of an up-down adjacency. Naturally, any logical matrix representation of an up-down relation is built as the power sum of the logical matrix representation of its corresponding up-down adjacency. Building the representation of an up-down relation costs $O(n^2)$ time because the up-down adjacency is expressed as a chain. For example, corresponding to the final state shown in Figure 5, the transitive closure of P_1 is { (1, 4), (4, 3), (3, 2), (1, 3), (4, 2), (1, 2)}. The corresponding logical matrix representation of this up-down relation is obtained through the following computation.

- The logical matrix representation of the up-down adjacency of P_1 is

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

- The matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

corresponds to ${P_1}^2 = \{(1, 3), (4, 2)\}$.

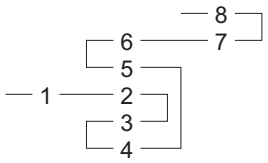

FIGURE 6. A middle state of a map of size 1×8.

- The matrix
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
corresponds to ${P_1}^3 = \{(1,2)\}$.

- Because when $i > 3$, ${P_1}^i$ degenerates to 0, we have the power sum matrix
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
which corresponds to
$$\begin{aligned} {P_1}^+ &= P_1 + {P_1}^2 + {P_1}^3 \\ &= \{(1,4),(4,3),(3,2),(1,3),(4,2),(1,2)\}. \end{aligned}$$

Logical matrix representation of middle states. We have discussed the logical matrix representations of final states in the above two paragraphs. They can also be extended to represent middle states by applying further consideration on whether the front side or the back side of a square faces up in a middle state. Without loss of generality, we assume that Square 1 will have its front side up in the final state.

Firstly, we add the notion *parity*. We always assume that the leftmost square in a map of size $1 \times n$ or the square at the bottom left of a map of size $m \times n$ is the *origin* to standardize the parity of any other squares and is with the *even parity* itself. We also assume that in any final state, the origin will always be frontside up. In a map of size $1 \times n$, we say the distance between the i-th square and origin is $i - 1$. In a map of size $m \times n$, we define the distance between the square on the i-th row and the j-th column and the origin as the Manhattan distance between them, i.e., $i+j-2$. We say the parity of the distance between a square and the origin is the parity of the square. By the conclusion of [3], a pair of adjacent squares a, b with $a < b$ in the middle state would be assigned $a < b$ in the logical matrix (the final state) in two cases. The first case is that the parity of a is odd and a is frontside down. The second case is that the parity of a is even, and a is frontside up in the middle state. Otherwise, a, b with $a < b$ in a middle state would be assigned $b < a$ in the corresponding logical matrix. This assignment follows the analysis in [3].

Take the middle state shown in Figure 6 as an example. In this middle state, all the adjacencies are included in $R = \{6 < 5, 5 < 2, 2 < 3, 3 < 4, 7 < 8\}$. $7 < 8$ follows the parity. Even though the practical overlapping of squares 2, 3, 4, 5, 6 is in reverse in

this middle state, we still assign the elements of the logical matrix following what they are supposed to be in the final state for the unity. We assign two neighbors an up-down adjacency, which only depends on the assignment of the crease between them.

The logical matrix of all the up-down adjacency in R is:

$$A_R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Computing its power sum, because ${A_R}^k$ becomes 0 when $k > 4$, we get:

$$E_R = \sum_{i=1}^{4} {A_R}^i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

E_R reveals all the up-down relations in this middle state.

In conclusion, we have the following theorem, which is straightforward according to the definition of the up-down adjacency and the up-down relation as its transitive closure.

Theorem 3.6. *There exists a one-to-one mapping from the set of legal flat middle or final states to the set of their matrix representations (either the representations of the up-down adjacency or the representation of the up-down relation).*

3.3. Logical Matrix Representation of $m \times n$ maps

As mentioned above, the logical matrix representation of $m \times n$ maps is adapted to a four-dimensional matrix representation. Even though other possible arrangements of these layers in our four-dimensional matrix representation can flatten it into lower dimensions, we here use the four-dimensional form so as to clearly represent the simple folds and then simple unfolds at particular contiguous positions in the matrix.

Compared to a $1 \times n$ map, both horizontal pairs of squares and vertical pairs of squares can intersect with their same type (horizontal or vertical) in an $m \times n$ map. For example, in an ordering of the squares in the 4×4 map as illustrated in Figure 7, a pair of squares $\{1, 2\}$ may intersect with $\{5, 6\}$, while it would never intersect with $\{9, 13\}$ [3]. Based on this conclusion, the basic structure of our matrix should provide an available check on the penetrations of horizontal pairs and the penetrations of vertical pairs. First, considering the horizontal pairs, every row of the map is viewed as a $1 \times n$ map, and the entire map is viewed as a combination of this $1 \times n$ map and $m - 1$ copies of it. Penetrations could happen either within this map (such as $\{1, 2\}$ and $\{3, 4\}$ in the instance in Figure 7) or between this map and its copies((such as $\{1, 2\}$ and $\{7, 8\}$ in the instance in Figure 7)). Either case corresponds to a two-dimensional $n \times n$ section of our four-dimensional matrix.

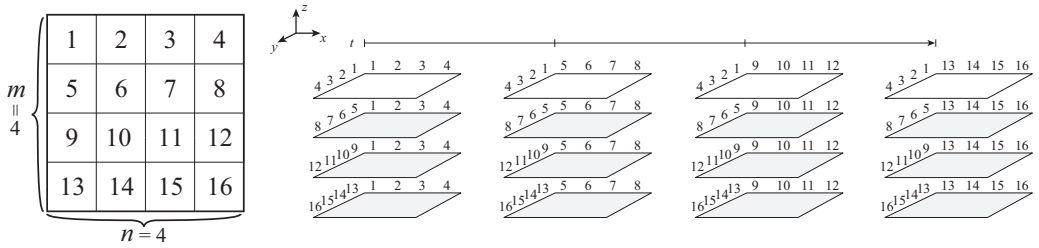

FIGURE 7. Four-dimensional matrix representation of a 4×4 map. The numbers are indices of squares. Every parallelogram indicates a 4×4 matrix.

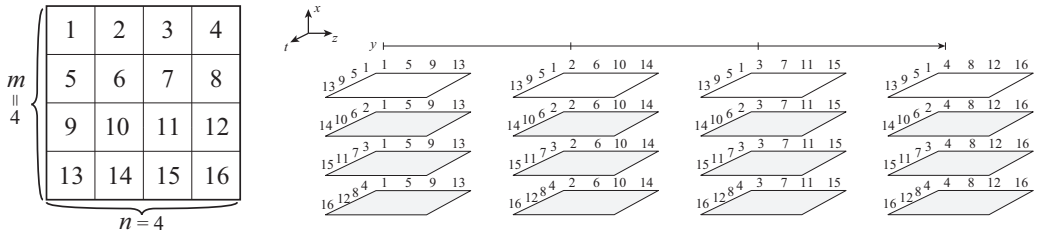

FIGURE 8. Another description of the four-dimensional matrix representation in Figure 7.

In this way, if we fix a row, a three-dimensional matrix comprised of all the $n \times n$ sections involving this row can be created. This three-dimensional matrix is sufficient for checking the intersection involving the row [3]. We then build such a three-dimensional matrix for every row, respectively. In this way, the four-dimensional matrix of a map of size $m \times n$ is established as an $m \times m \times n \times n$ matrix, which can be flattened to form a two-dimensional $mn \times mn$ matrix.

Precisely, the above construction can be detailed as: for any $m \times n$ map with m rows and n columns. We define a four-dimensional matrix M with its x-axis and y-axis representing the rows and its z-axis and t-axis representing the columns. For example, for a 4×4 map on the left side of Figure 7, the structure of its corresponding matrix representation is specified as the four-dimensional matrix that projects to the four three-dimensional matrices on the right side of Figure 7. Every layer in the figure, representing a 4×4 matrix, is parallel to the xy-plane. The numbers here are indices of squares. If we join the layers with contiguous indices, we can get a 16×16 matrix with both its row and column indicating the 16 squares in the map.

Next, for all the columns, the same check is supposed to be done as what we analyzed for the rows. We should check the penetrations involving pairs of vertical neighbor squares. This time, we also view each column of the $m \times n$ map as an $m \times 1$ map. By fixing one column and viewing other columns as its copies, we can build three-dimensional matrices with n layers, each specified as a $m \times m$ matrix. Considering all the columns, we finally obtain a four-dimensional matrix comprising n such three-dimensional matrices. If we

project the four-dimensional matrix representation we constructed for the rows to either the xzt- or yzt-section, then the resulting three-dimensional matrices exactly reflect the penetrations of vertical neighbor squares. In Figure 8, we present an example of the xzt-projection of the structure given in Figure 7. Every layer in this figure represents a vertical 4×1 map.

3.4. Determine the Relationship between Accessible Folded States of Different Fold Models according to the Matrix Representation

This part describes how to use the four-dimensional matrix representation to determine the relationship between the accessible folded states of the above-mentioned three distinct fold models. First, we can claim Theorem 3.7.

Theorem 3.7. *All the accessible flat-folded states of the simple fold model can be accessed by the simple fold-unfold model. All the accessible flat-folded states of the simple fold-unfold model can be accessed by the general fold model.*

Proof. This conclusion straightly follows the definition of these fold models. ∎

In the above paragraphs, we have shown the uniqueness of the correspondence between the representation of the up-down adjacency and the representation of the up-down relation, as the latter could be obtained as a power sum of the former.

To handle an $m \times n$ map, we can transform the four-dimensional matrix into a two-dimensional $mn \times mn$ matrix while preserving the correspondence between elements and their row and column indices. This is accomplished by numbering all the squares and using the same logical matrix representation method as in the case of a $1 \times n$ map to create the two-dimensional $mn \times mn$ matrix. The row indices and column indices of the two-dimensional matrix are supposed to be assigned with values ranging from 1 to mn, which indicate the corresponding squares. Its power sum matrix captures the information of the up-down relation in this manner. Based on this fact, to discuss how different kinds of folds influence the overlapping order, it is sufficient to discuss what corresponds to different kinds of folds in a logical matrix representation of the up-down adjacency. In other words, we should figure out how the logical matrix representation of the up-down adjacency changes when different folds occur.

We have the following theorem for general folds because its only constraint is no penetration.

Theorem 3.8. *The representation of general fold is legal as long as in its corresponding matrix representation of the up-down relation, in any 2×2 sub-matrix representing two pairs of neighbor squares with the same parity, there are an even number of elements assigned 1. Here, seeing the projection of the map to the x-axis or the y-axis as a $1 \times n$ map, the parity means that the two pairs of neighbor squares correspond to either $\{(2i, 2i+1), (2j, 2j+1)\}$, or $\{(2i-1, 2i), (2j-1, 2j)\}$ in the projection $1 \times n$ map (where both i and j are integers). General folds change elements of the representation of the up-down adjacency separately, i.e., they can change discontiguous matrix elements.*

Proof. First, penetration occurs when a pair of neighbor squares has exactly one square sandwiched between the other pair of neighbor squares. Based on a conclusion of [11], in the projection $1 \times n$ map, only two kinds of pairs: $\{(2i, 2i+1), (2j, 2j+1)\}$ and $\{(2i-1, 2i), (2j-1, 2j)\}$ can bring about penetrations. Without loss of generality, we analyze the penetration occurring between a pair $\{(2i, 2i+1), (2j, 2j+1)\}$. Assuming that $2i$

is sandwiched between $2j$ and $2j+1$, then in the corresponding projection of the four-dimensional matrix, which forms a two-dimensional matrix M itself, $m_{2i,2j}$ and $m_{2i,2j+1}$ would be assigned 1 and 0 or 0 and 1, respectively. While for $2i+1$, which is either below or over the whole pair $\{2j, 2j+1\}$, the corresponding $m_{2i+1,2j}$ and $m_{2i+1,2j+1}$ would be assigned both 1 or both 0. Therefore, in the representation of a legal state, there should be an even number of elements assigned 1.
Second, general folds change matrix elements separately because we can adjust the position of only one square to transform a final state to another final state. Here, we only consider the up-down adjacency matrix. We still analyze this conclusion in the projection matrix and see the projected map as a $1 \times n$ map. Suppose a square s is moved from an overlapping order $(\cdots, r_1, s, t_1, \cdots)$ to $(\cdots, r_2, s, t_2, \cdots)$, then, in the corresponding projection two-dimensional matrix, only up to four elements (some of r_1, r_2, t_1 and t_2 may not exist), $m_{r_1,s}$, m_{s,t_1} and $m_{r_2,s}$, m_{s,t_2} would be altered. As an example, when we want to transform the final state (b) to (c) in Figure 5, we only have to adjust the position of square 1, and only $m_{1,3}$, $m_{2,1}$, and $m_{4,1}$ are supposed to be altered. ∎

We wish to prove that the sets of accessible final states of the simple fold-unfold model and the simple fold model are both properly included in the set of accessible final states of the general fold model. To prove this, it is sufficient to consider the first simple (un)fold applied on the map to see how stricter the constraint is than the general fold model. The reason is that every simple (un)fold has exactly the same property if the current middle state is viewed as a reduced map. Based on this consideration, we have the following theorem for the first simple fold and its reverse unfold.

Theorem 3.9. *The first simple fold and its reverse unfold must influence (alternately change) the diagonal elements of certain layers parallel to either the xy-plane or the zt-plane in the four-dimensional matrix.*

Proof. According to the definitions, any simple fold or simple unfold influences the adjacency between pairs of squares at symmetric positions with respect to the line segment along which the fold has been implemented. In other words, at least all the adjacency between every pair of squares in the two rows or two columns with the folded line as the center-line are supposed to be altered simultaneously. In the corresponding matrix representation, we use x, y to represent column (to index columns) and use z, t to represent rows (to index rows). Correspondingly, the alteration is represented as the alteration of all the diagonal elements of certain layers parallel to either the xy-plane or the zt-plane. ∎

For example, the simple fold of a 4×4 map as illustrated on the upper side of Figure 9 influences the diagonals of two layers along the zt-plane, as the ones colored grey on the lower side of Figure 9. These two layers in the third matrix and the fourth matrix are respectively assigned as:

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

representing the adjacencies $\{3<4, 8<7, 11<12, 16<15\}$.

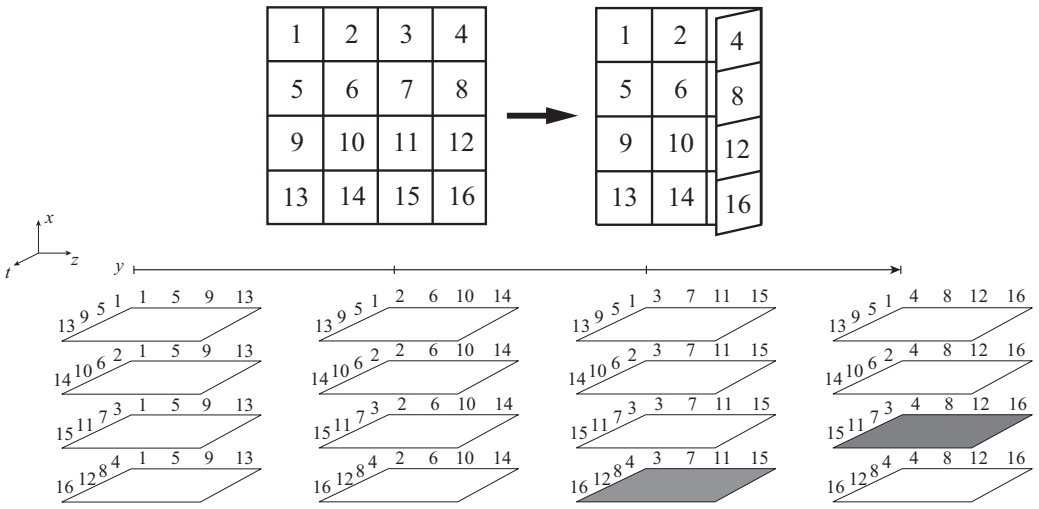

FIGURE 9. A simple fold and its corresponding assignment in the four-dimensional matrix representation. The numbers are indices of squares.

Because in VTP, all the squares are concerned, Theorem 3.9 indicates that the restriction on the alteration induced by a simple fold is stricter than the general fold. In other words, the set of accessible total overlapping orders of the simple fold-unfold model is properly included in the set of accessible total overlapping orders of the general fold model. Then, to discuss VBP, we give the following theorem.

Theorem 3.10. *The effect of the first simple fold and its reverse unfold on the boundary squares is the two end elements of each assigned diagonal discussed in the last theorem.*

Proof. This conclusion is straight-forward according to the definition of VBP. ∎

The fact that both ends are affected simultaneously tells that even in VBP, the set of accessible boundary overlapping orders of the simple fold-unfold model is still properly included in the set of accessible boundary overlapping orders of the general fold model.

Finally, we claim Theorem 3.11 to differentiate the accessible overlapping orders of the simple fold-unfold model and the simple fold model.

Theorem 3.11. *In a folding process, horizontal folds and vertical folds do not commute in general.*

Proof. This conclusion is apparent because once a horizontal fold and a vertical fold have an intersection, the overlapping order involving the square at their intersection can be different if their order is changed, as illustrated in Figure 10. Assuming that b would have its front side up in the final state, the overlapping order of squares a, b, c would be (b, c, a) in the left figure and (b, a, c) in the right figure. ∎

Based on Theorem 3.11, we can further obtain the following theorem.

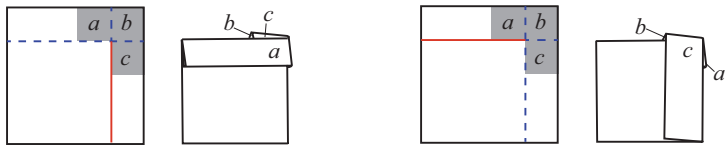

FIGURE 10. A horizontal fold and a vertical fold which do not commute. The order of the folds is (horizontal, vertical) in the left figure and is the reverse in the right figure.

Theorem 3.12. *If there exist middle states of a* $1 \times n$ *map which are accessible by the simple fold-unfold model but not the simple fold model, then there exist final states of an* $m \times n$ *map which are accessible by the simple fold-unfold model but not the simple fold model.*

Proof. We assume that a middle state of a $1 \times n$ map is accessible by the simple fold-unfold model but not the simple fold model and that it is accessed by a sequence of simple folds and unfolds S_1. Then, a final state accessible by the simple fold-unfold model but not by the simple fold model can be obtained by: first folding an $m \times n$ map with vertical folds the same as S_1, then applying some arbitrary horizontal folds (denoted S_2), and finally folding the map to size 1×1. The result overlapping order is inaccessible by a simple fold model because elements in S_1 and S_2 are not commutable according to Theorem 3.11.

Indeed there are middle states of a $1 \times n$ map that can be accessed by the simple fold-unfold model but not the simple fold model. A family of such states is illustrated in Figure 11. Based on the definition of the simple fold, we know that if a middle state is accessible by the simple fold model, then there must be a sequence of only simple unfolds to unfold such a state to a piece of paper. In this figure, because both ends are sandwiched between other continuous layers, unfolds involving them cannot be applied. Then, the key part keeping it from being accessed by the simple fold model is its left end. Only when this part does not exist, i.e., when the two creases respectively denoted V and M align on the same vertical line, the two creases can be unfolded together and lease an end of the map from the sandwiched state. Therefore, this family cannot be accessed by the simple fold model. On the other hand, [4] showed that every legal middle state, including this family, is accessible by the simple fold-unfold model (which was referred to as the simple fold model in their terminology). Therefore, the set of accessible total overlapping orders by the simple fold model is properly included in the set of accessible total overlapping orders by the simple fold-unfold model. When m is even, we specify S_2 as the fold along the horizontal centerline of the $m \times n$ map. Then, clearly, the resulting boundary overlapping order is inaccessible by only simple folds. When m is odd, we specify S_2 as a sequence of two folds, one along the $(m-1)$-st horizontal line (counting from either the top or the bottom), which will reduce the map to a middle state with $m-1$ rows, then, the other fold is supposed to be done along the horizontal centerline of the middle state (illustrated in Figure 12). The result boundary overlapping order would also be inaccessible by only simple folds. ∎

In conclusion, we have the following theorem.

Theorem 3.13. *The relationships between different fold models illustrated in Table 1 hold.*

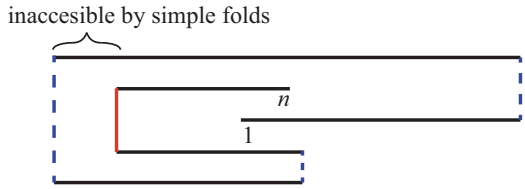

FIGURE 11. A middle state of a $1 \times n$ map which can be accessed by the simple fold-unfold model but not the simple fold model.

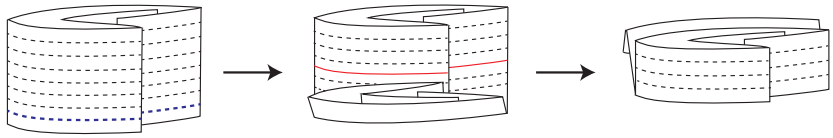

FIGURE 12. How to construct a boundary order inaccessible by the simple fold model.

4. Verification by Examples

In this section, we give examples from the relative complements to verify our conclusion in Section 3.

Inclusion relationship between the final states of the simple fold-unfold fold model and the general fold model. We start the discussion from VTP instances. An element of the relative complement of the accessible overlapping order of the general fold model and the accessible overlapping order of the simple fold-unfold model, i.e., an order accessible to the general fold model but inaccessible to either the simple fold-unfold model or the simple fold model, is provided in Figure 13. The total overlapping order on the left side was first proposed in [8].

This instance also directly leads to a desired VBP instance. Its corresponding boundary overlapping order is given on the right side of Figure 13. Particularly, in this instance, only the center square labeled five in the left figure is not on the boundary, while the order formed by the others uniquely decides the MV assignment. Because there exists no group of creases lying on the same line that all assigned valleys or mountains, no simple fold could be applied in the first place [8]. Therefore, it provides a boundary order which the general fold model can access while neither the simple fold-unfold model nor the simple fold model can access.

Inclusion relationship between the final states of the simple fold model and the simple fold-unfold model. This time, we provide an instance that satisfies both the setting of VTP and the setting of VBP, as illustrated on the left side of Figure 14. Clearly, this order is both a total and a boundary order. To further explain its accessibility and inaccessibility with respect to the simple fold-unfold model and the simple fold model, we draw the side view of the corresponding final state on its right side with the number 1 representing the side length of a square. The accessibility and inaccessibility can easily be tested by hand.

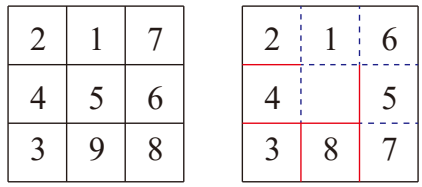

FIGURE 13. A boundary overlapping order that can be accessed by general folds but not only simple folds and simple unfolds.

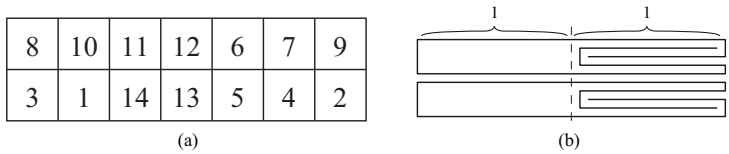

FIGURE 14. An instance of both VTP and VBP that the simple fold-unfold model can access while the simple fold model cannot.

5. Conclusion and Future Work

In this study, we presented a new logical matrix representation of map folding, based on which we further investigated the relationships of three commonly used fold models by a four-dimensional logical matrix representation of an $m \times n$ map. We theoretically proved the proper inclusion relation between the sets of accessible final states of every two fold models in VTP and VBP, respectively. The logical matrix representation visualizes the performance of different types of folds and the difference between them clearly and precisely. As future work, we wish to explore the potential for this matrix representation to be employed in many other contexts, including both problem-based computational investigations of origami (e.g., solving the original map folding problem and improving the solution in the case that the size of the map is $2 \times n$.) and the essential association between the field of origami and other mathematical fields. For example, from a mathematical viewpoint, depending on the semi-ring algebraic structure and its natural association with monoidal categories, the matrix representation is supposed to be developed as a powerful medium for studying foldings and fold models via their mathematical properties.

References

[1] M. Bern, B. Hayes, The complexity of flat origami, Proc. 7th Ann. ACM-SIAM Symp. on Discrete Algorithms, ACM (1996) 175–183.

[2] H.A. Akitaya, K.C. Cheung, E.D. Demaine, T. Horiyama, T.C. Hull, J.S. Ku, R. Uehara, Box pleating is hard, Japanese Conference on Discrete and Computational Geometry and Graphs, Springer, Cham (2015) 167–179.

[3] R.I. Nishat, Map Folding, Master's Thesis, Bangladesh University of Engineering and Technology, Bangladesh, 2009.

[4] R. Uehara, Stamp foldings with a given mountain-valley assignment, AK Peters/CRC Press; Origami 5 (2016) 599–612.

[5] Y. Jia, J. Mitani, R. Uehara, Research on map folding with boundary order on simple fold, IEICE Trans. Fundamentals. E104 (A. 9) (2021) 1116–1126.

[6] Y. Jia, J. Mitani, A comparison of different fold models in variations of the map folding problem, Applied Sciences 11 (24) (2021) 11856.

[7] Y. Jia, Matrix representations of one-dimensional map folding and single-vertex flat folding, 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), IEEE (2021) 307–311.

[8] E.D. Demaine, J. O'Rourke, Geometric Folding Algorithms: Linkages, Origami, Polyhedral, 1st ed., Cambridge University Press: Cambridge, UK, 2007.

[9] Y. Jia, J. Mitani, R. Uehara, Valid orderings of layers when simple-folding a map, Journal of Information Processing 28 (2020) 816–824.

[10] D.J. Lehmann, Algebraic structures for transitive closure, Theoretical Computer Science 4 (1) (1977) 59–76.

[11] J.E. Koehler, Folding a strip of stamps, Journal of Combinatorial Theory 5 (2) (1968) 135–152.

[12] C. Heunen, J. Vicary, Categories for Quantum Theory: An introduction, Oxford University Press, 2019.

[13] S. Awodey, Category Theory, Oxford University Press, 2010.

[14] A. Grothendieck, Sur quelques points d'algèbre homologique, Tohoku Mathematical Journal, Second Series 9 (2) (1957) 119–183.

[15] G.E. Bredon, Sheaf Theory, Springer Science and Business Media, Vol. 170, 2012.