The Mathematics of Origami

Thomas Bertschinger, Joseph Slote, Claire Spencer, & Samuel Vinitsky

Carleton College

February 22, 2016

Overview

Axioms of Origami

General

i. Brief History

ii. Independence

Axioms of Origami

Seven Axioms of Origami

i. Given two points p_1 and p_2, we can fold a line connecting them.
ii. Given two points p_1 and p_2, we can fold p_1 onto p_2.
iii. Given two lines l_1 and l_2, we can fold line l_1 onto l_2.
iv. Given a point p_1 and a line l_1, we can make a fold perpendicular to l_1 passing through the point p_1.

Axioms of Origami

Seven Axioms of Origami

v. Given two points p_1 and p_2 and a line l_1, we can make a fold that places p_1 onto l_1 and passes through the point p_2.

vi. Given two points p_1 and p_2 and two lines l_1 and l_2, we can make a fold that places p1 onto line l_1 and places p_2 onto line l_2.

vii. Given a point p_1 and two lines l_1 and l_2, we can make a fold perpendicular to l_2 that places p_1 onto line l_1.

Axioms of Origami

Seven Axioms of Origami

i. The first six axioms allow all quadratic and cubic equations with rational coefficients to be solved.

ii. They also allow two of the three problems of antiquity, the trisection of an angle and the doubling of the cube, to be constructed.

Greek Problems of Antiquity

Problems of Antiquity

These were a trio of geometric problems whose solutions were attempted solely through the use of compass and straight-edge.

i. Angle Trisection
ii. Cube Duplication
iii. Circle Squaring

Angle Trisection

INCLUDE PARTIALLY FOLDED DIAGRAMS

Cube Duplication

INCLUDE PARTIALLY FOLDED DIAGRAMS

i. $\frac{\alpha}{\beta} = \sqrt[3]{2}$
ii. Thus, a cube with a side length α will have twice the volume of a cube with side length β.

Squaring the Circle

i. Impossible
ii. π

Constructability

Constructible Numbers

i. Given two points p_0 and p_1, construct a third point p_1' a distance $|p_0 p_1|$ from point p_0 such that $\overline{p_0 p_1'}$ is perpendicular to $\overline{p_0 p_1}$.

ii. Given two points p_0 and p_1, a third point p_2 can be constructed such that p_2 is collinear with p_0 and p_1, thus $|p_0 p_1| = |p_1 p_2|$.

Constructability

Constructible Numbers

iii. Given two constructible numbers α and β, we can construct $\frac{\alpha}{\beta}$, their ratio.

iv. Given two constructible numbers α and β, we can construct their sum $\alpha + \beta$ or their difference $\alpha - \beta$.

v. Given two constructible numbers α and β, we can construct $\alpha\beta$, their product.

Constructability

Given two constructible numbers α and β, we can construct $\frac{\alpha}{\beta}$, their ratio.
INCLUDE PARTIALLY FOLDED DIAGRAMS AS WELL AS FINAL FOLDED DIAGRAM

Constructability

Constructible Numbers

i. Since the set of constructible numbers is closed under addition, subtraction, multiplication, and division, it can be concluded that the set of constructible numbers form a field.

ii. Ultimately, the field of origami constructible numbers are closed under taking both square roots and cube roots.

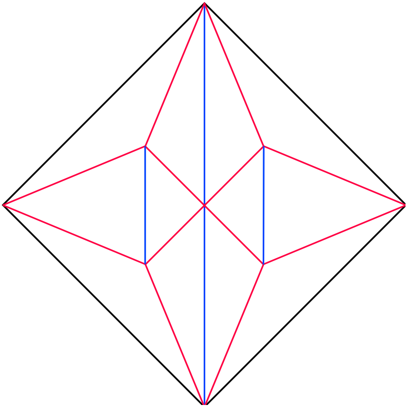
iii. The construction of the square root of any constructible number implies the field of origami constructible numbers contains the field of compass and straightedge constructible numbers.

Future Work

i. Efficiency
ii. Optimality

Foldability

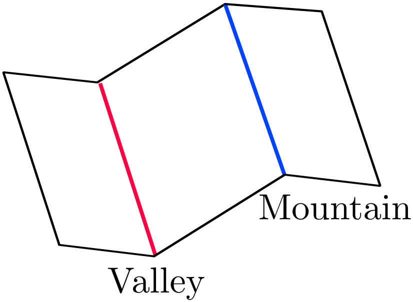
An example of a crease pattern:



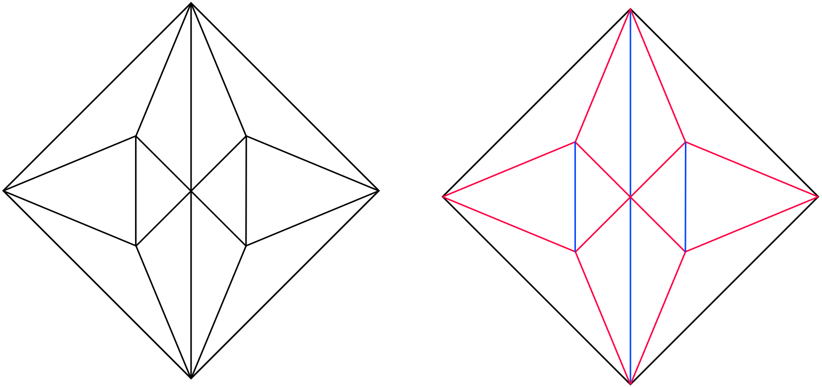
Foldability

Crease Assignments

A crease pattern doesn't contain all the information about a model, however.



Foldability



Foldability

Two questions

For a given crease pattern,

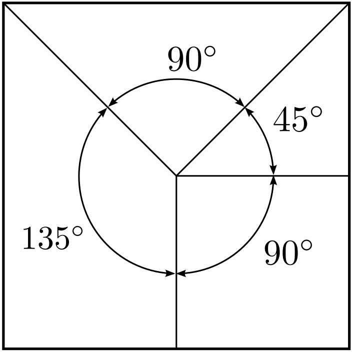
i. is there a way to fold it flat?

ii. is there a way to fold it at all?

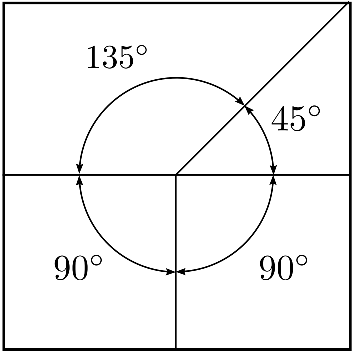
Flat-Foldability Conditions

Kawasaki's Theorem (1989)

The alternate angles around a vertex must sum to 180°.

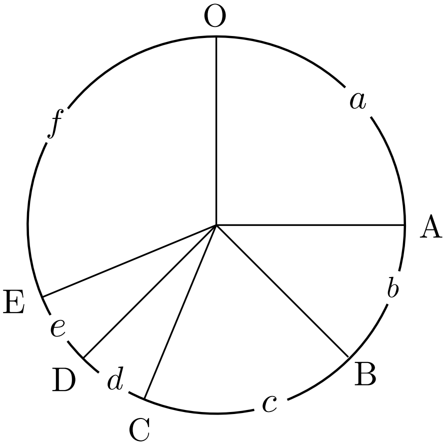


(a)

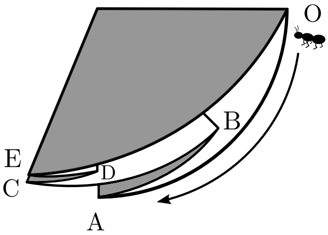


(b)

Kawasaki's Theorem



(a)

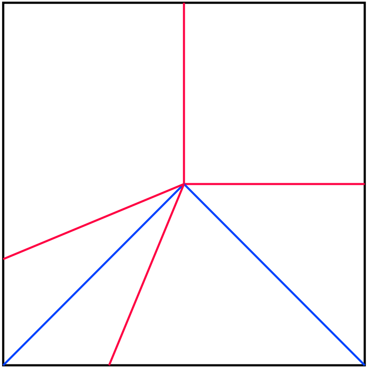


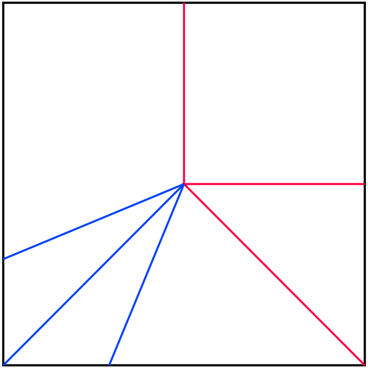
(b)

Flat-Foldability Conditions

Maekawa's Theorem (1986)

The sum of mountain + valley is ± 2.



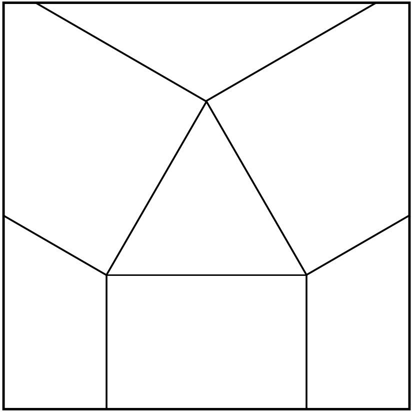


More Complicated Crease Patterns

How can we determine flat foldability for a more complicated pattern?

More Complicated Crease Patterns

Here's a crease pattern that can't fold flat!

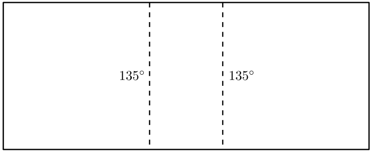


Foldings

How can we capture the notion of a folded paper?

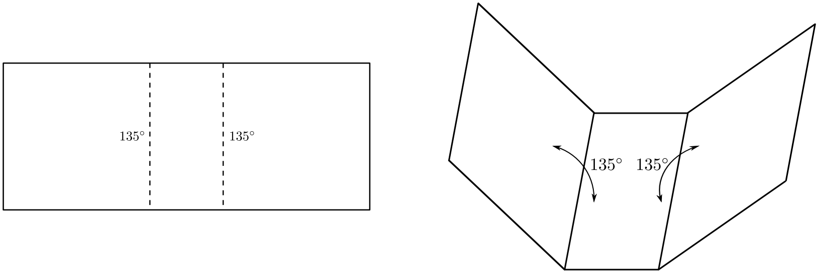
Foldings

How can we capture the notion of a folded paper?



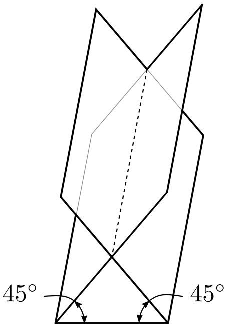
Foldings

How can we capture the notion of a folded paper?



An Invalid Folding

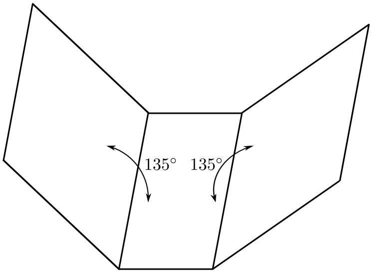
This folding has self-intersection.



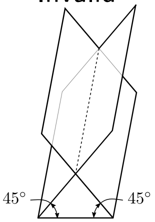
Valid Foldings

A valid folding is one that doesn't cause any self-intersecton.

Valid



Invalid



Drawing pictures on Crease Patterns

Question

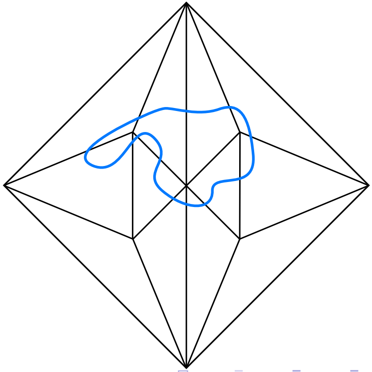
If we draw something on the flat piece of paper before folding it up, what can happen when we fold it?

Drawing pictures on Crease Patterns

Question

If we draw something on the flat piece of paper before folding it up, what can happen when we fold it?

We'll see what happens when we draw Jordan curves on the crease patterns, like the picture on the right.

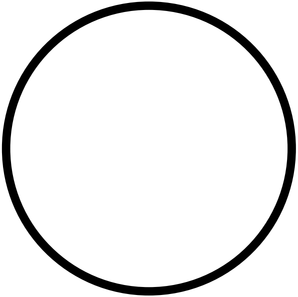


Knot Theory 101

What is a knot?

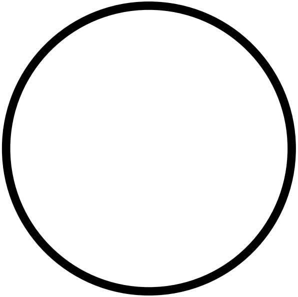
Knot Theory 101

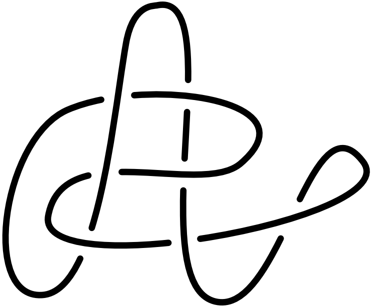
What is a knot?



Knot Theory 101

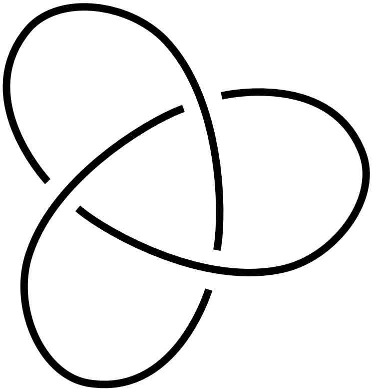
What is a knot?





Knot Theory 101

This is the *trefoil knot*. We can't "untangle" it, no matter how hard we try.



Connecting Topology and Origami

Theorem

A folding of a crease pattern C is valid if and only if every Jordan curve embedded in the paper before folding is mapped to the unknot after folding.

Connecting Topology and Origami

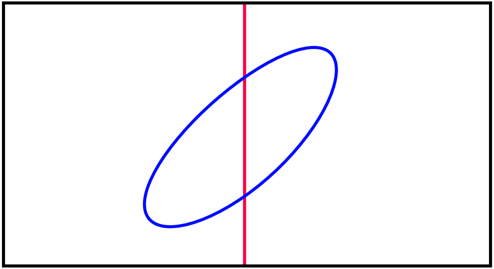
$\Longrightarrow$

This direction is fairly intuitive.

Connecting Topology and Origami

$\Longrightarrow$

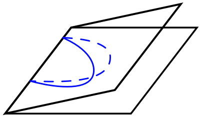
This direction is fairly intuitive.



Connecting Topology and Origami

$\Longrightarrow$

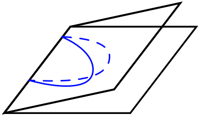
This direction is fairly intuitive.

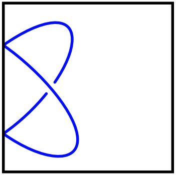


Connecting Topology and Origami

$\Longrightarrow$

This direction is fairly intuitive.

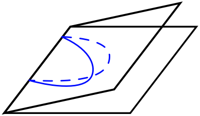
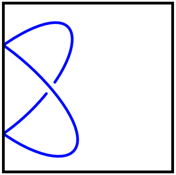
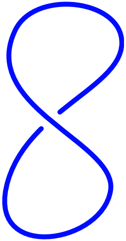




Connecting Topology and Origami

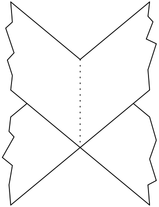
$\Longrightarrow$

This direction is fairly intuitive.

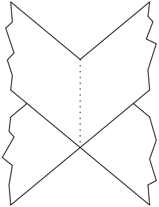
Connecting Topology and Origami

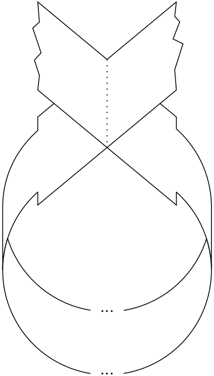
Invalid folding means intersection.



Connecting Topology and Origami

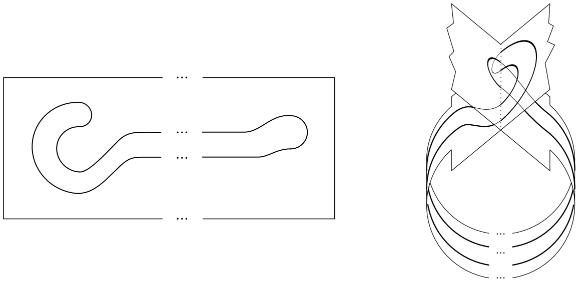
We can connect up the paper like this.





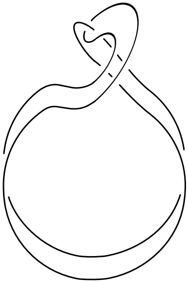
Connecting Topology and Origami

We can draw a curve like this.



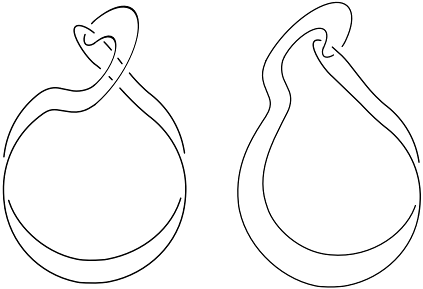
Connecting Topology and Origami

Is it a knot?



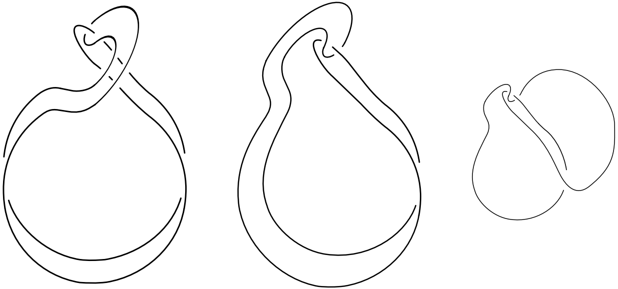
Connecting Topology and Origami

Is it a knot?



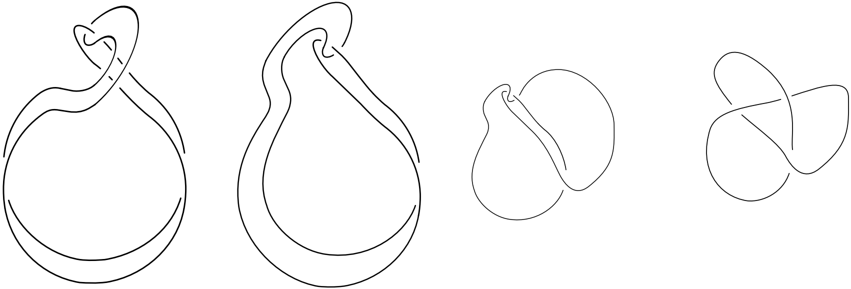
Connecting Topology and Origami

Is it a knot?



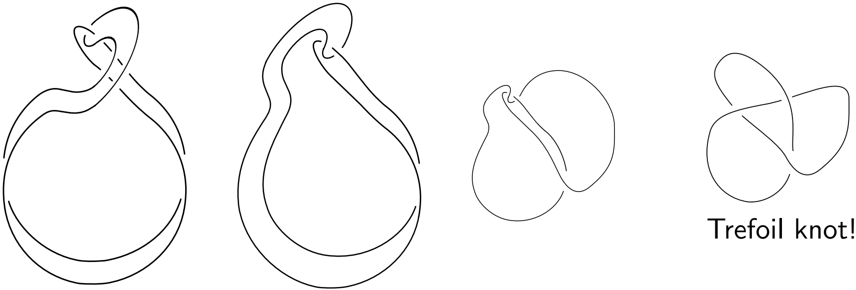
Connecting Topology and Origami

Is it a knot?



Connecting Topology and Origami

Is it a knot?



Trefoil knot!

Connecting Topology and Origami

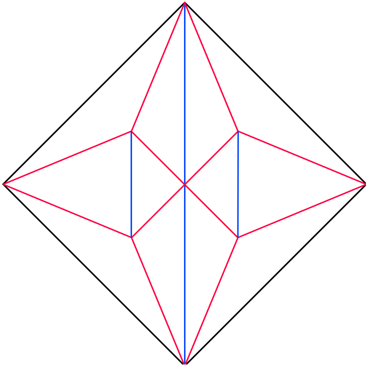
Now what?

- Maybe a characteristic for crease patterns
- Finding ways to describe the complexity of a crease pattern or a folding (e.g. the number of self-intersections)

Determining flat-foldability

Question:

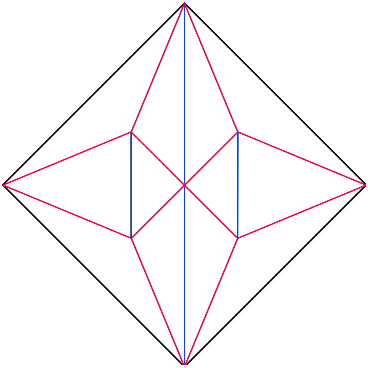
Is this crease pattern flat-foldable?

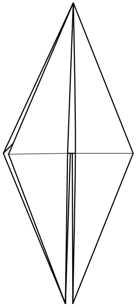


Determining flat-foldability

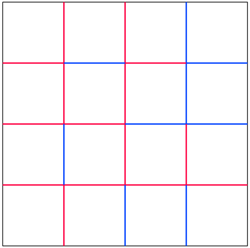
Question:

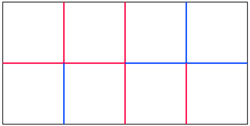
Is this crease pattern flat-foldable?



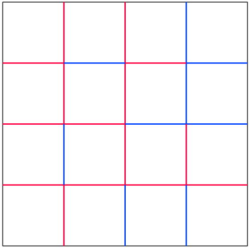


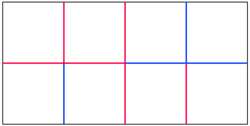
Introduction to Maps

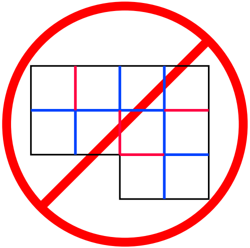


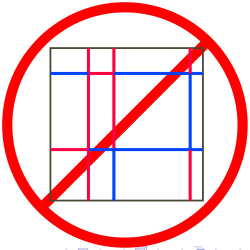


Introduction to Maps

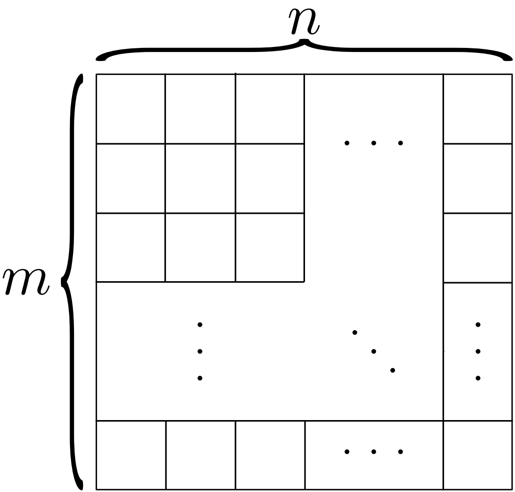








Introduction to Maps



Map Folding: An Open Problem

Open Problem:

How hard is it to determine whether or not a map is flat-foldable?

Map Folding: An Open Problem

Open Problem:

How hard is it to determine whether or not a map is flat-foldable?

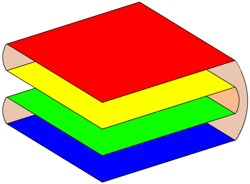
Easier Problem:

How hard is it to determine whether or not a map is *flat-folded*?

Linear Orderings

Question:

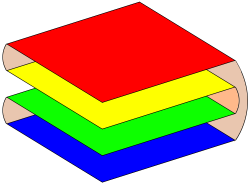
How can we represent a folded form?



Linear Orderings

Question:

How can we represent a folded form?

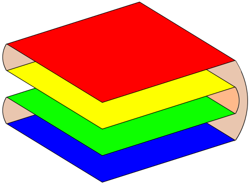


red $\rightarrow$ yellow $\rightarrow$ green $\rightarrow$ blue

Linear Orderings

Question:

How can we represent a folded form?



red $\rightarrow$ yellow $\rightarrow$ green $\rightarrow$ blue

Note:

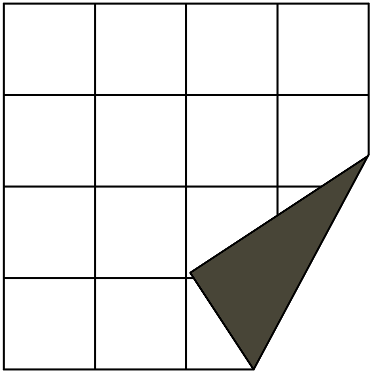
Each linear ordering corresponds with exactly one folded form.

Linear Orderings

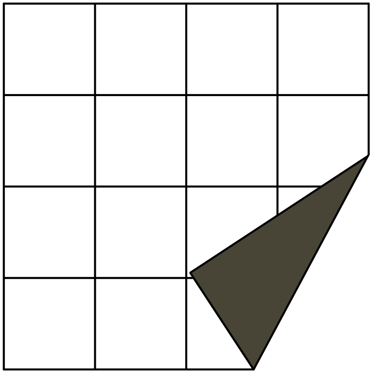
Question:

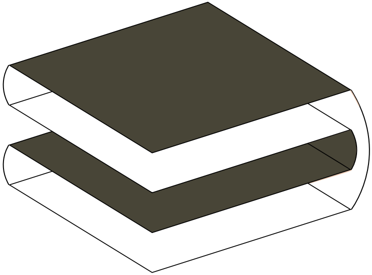
How can we tell if a linear ordering is realizable by a given crease pattern?

Checkerboard Pattern



Checkerboard Pattern





Checkerboard Pattern

Goal:

Label each face as being light-side or dark-side up in *any* folding.

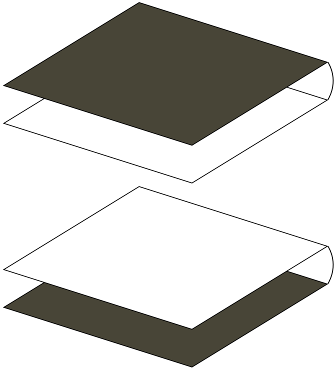
	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

Checkerboard Pattern

Goal:

Label each face as being light-side or dark-side up in *any* folding.

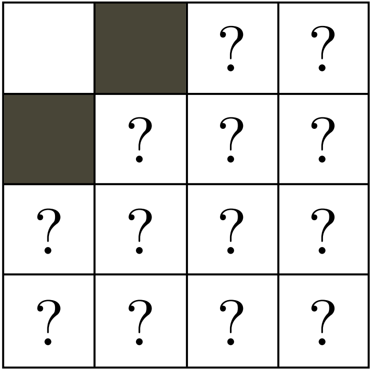
	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

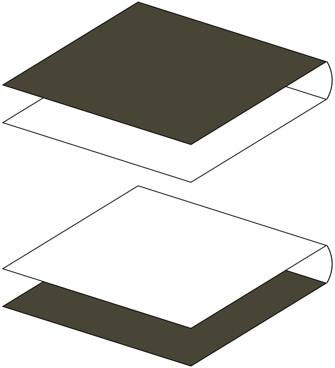


Checkerboard Pattern

Goal:

Label each face as being light-side or dark-side up in *any* folding.

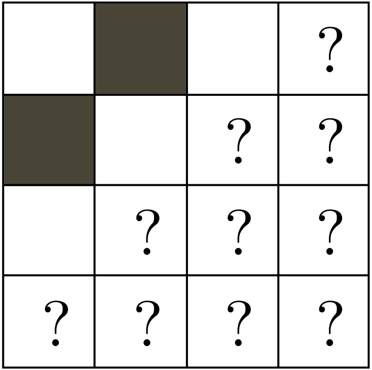


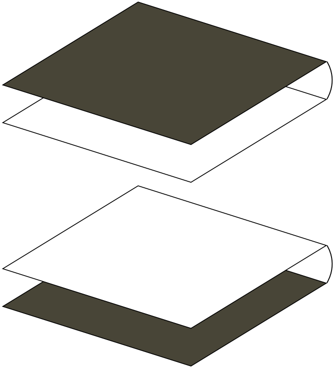


Checkerboard Pattern

Goal:

Label each face as being light-side or dark-side up in *any* folding.

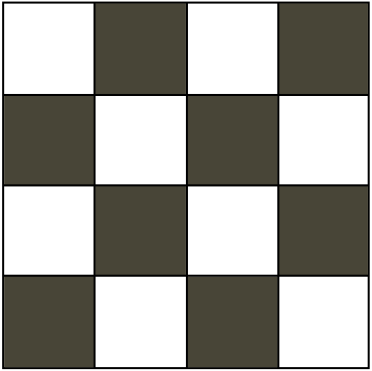


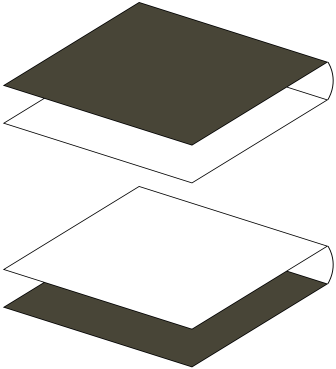


Checkerboard Pattern

Goal:

Label each face as being light-side or dark-side up in *any* folding.





Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

A partial ordering P on S might be $\{(a \rightarrow b), (c \rightarrow d)\}$.

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

A partial ordering P on S might be $\{(a \to b), (c \to d)\}$.

In every linear ordering that satisfies P, we know $a \to b$ and $c \to d$.

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

A partial ordering P on S might be $\{(a \to b), (c \to d)\}$.

In every linear ordering that satisfies P, we know $a \to b$ and $c \to d$.

Note however that there are many such linear orderings:

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

A partial ordering P on S might be $\{(a \rightarrow b), (c \rightarrow d)\}$.

In every linear ordering that satisfies P, we know $a \rightarrow b$ and $c \rightarrow d$.

Note however that there are many such linear orderings:

$a \rightarrow b \rightarrow c \rightarrow d$

Partial Ordering

Definition:

A *partial ordering* P on a set S is a set of ordered pairs of elements of S that orders some of the elements of S.

Example:

$S = \{a, b, c, d\}$

A partial ordering P on S might be $\{(a \rightarrow b), (c \rightarrow d)\}$.

In every linear ordering that satisfies P, we know $a \rightarrow b$ and $c \rightarrow d$.

Note however that there are many such linear orderings:

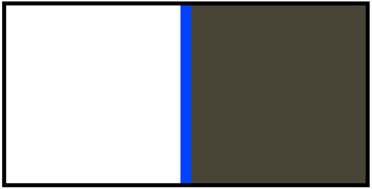
$a \rightarrow b \rightarrow c \rightarrow d$
$a \rightarrow c \rightarrow b \rightarrow d$
$c \rightarrow d \rightarrow a \rightarrow b$

...

Partial Ordering

Goal:

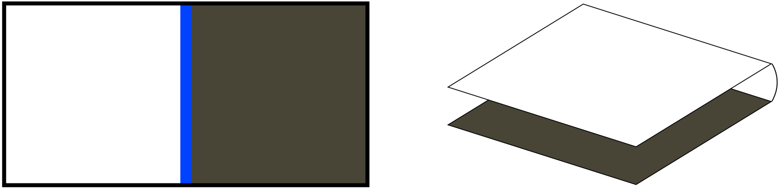
Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.



Partial Ordering

Goal:

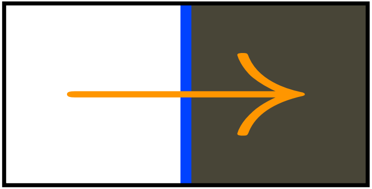
Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

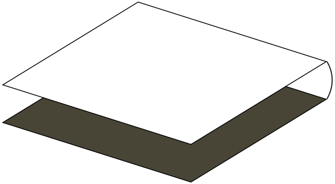


Partial Ordering

Goal:

Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

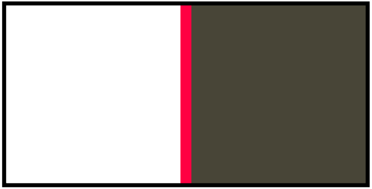




Partial Ordering

Goal:

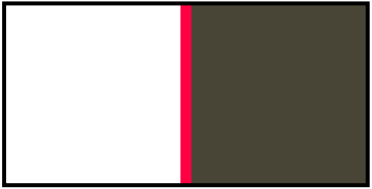
Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

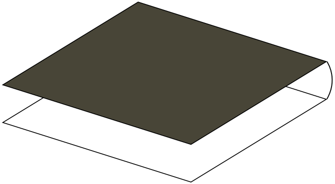


Partial Ordering

Goal:

Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

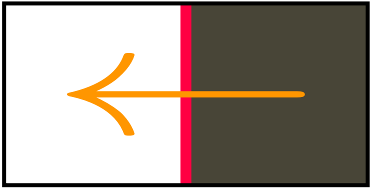


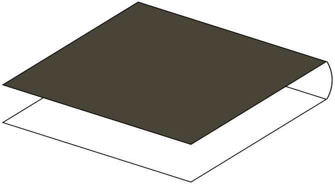


Partial Ordering

Goal:

Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

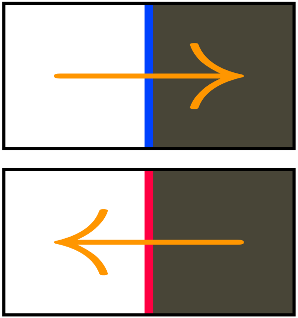




Partial Ordering

Goal:

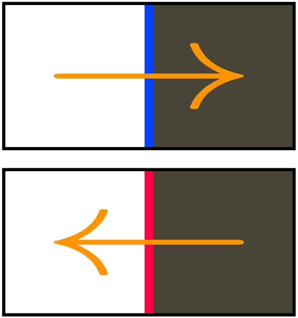
Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

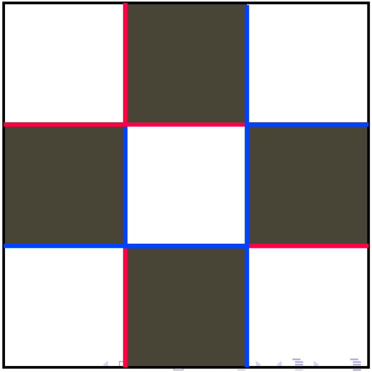


Partial Ordering

Goal:

Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.

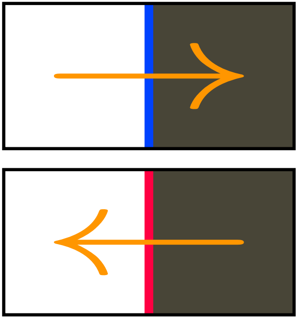


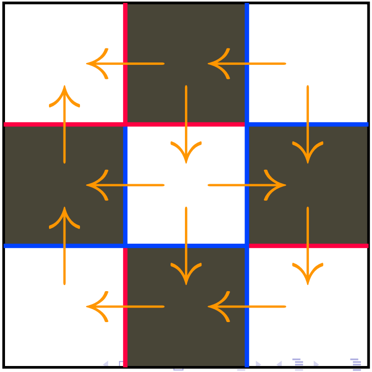


Partial Ordering

Goal:

Use checkerboard pattern and crease assignments to tell, for each pair of adjacent faces, which face comes first in *all* linear orderings.





Linear Orderings

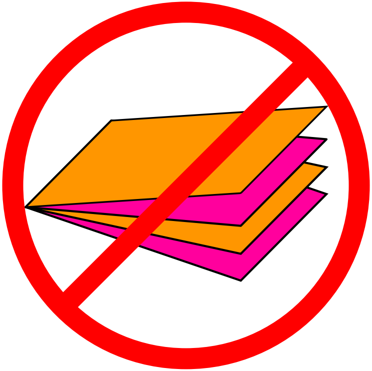
Is satisfying this partial ordering enough to ensure foldability?

Linear Orderings

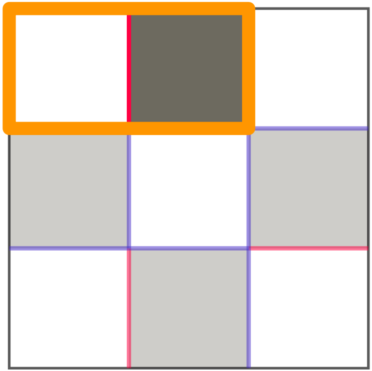
Is satisfying this partial ordering enough to ensure foldability? **No!**

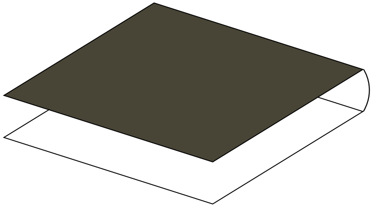
Linear Orderings

Is satisfying this partial ordering enough to ensure foldability? **No!**

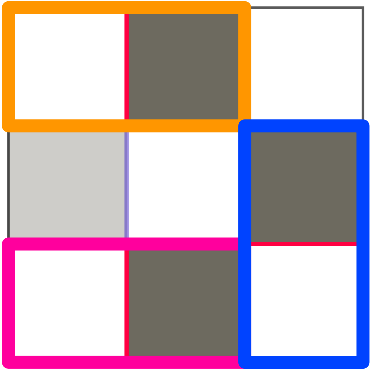


Butterflies

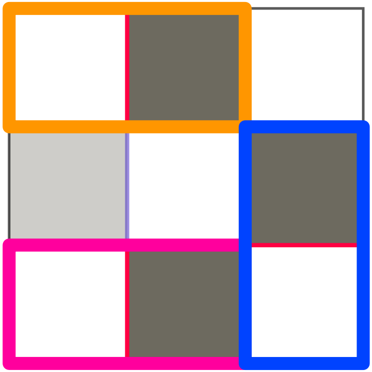


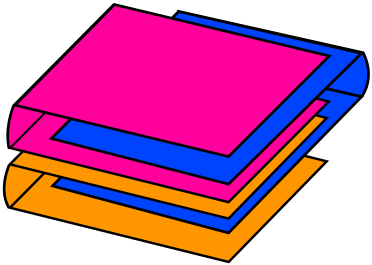


Butterflies



Butterflies

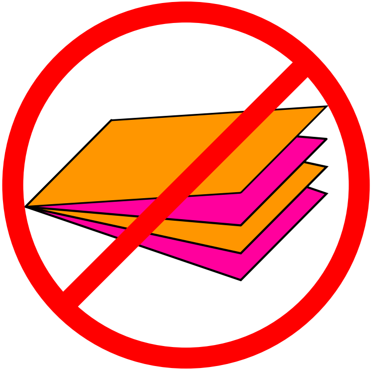




Butterfly Condition

Goal:

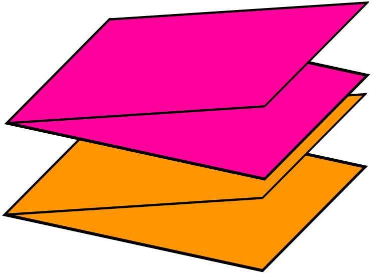
Enumerate the realizable configurations of twin butterfly pairs.



Butterfly Condition

Goal:

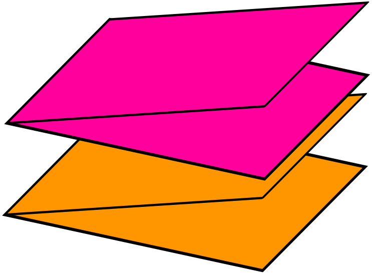
Enumerate the realizable configurations of twin butterfly pairs.

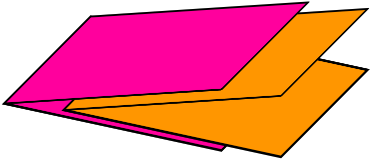


Butterfly Condition

Goal:

Enumerate the realizable configurations of twin butterfly pairs.





Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

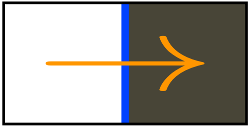
Proof: [$\Longrightarrow$] We have already proven this direction.

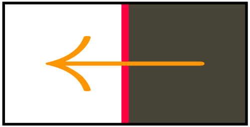
Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

Proof: [$\Longrightarrow$] We have already proven this direction.



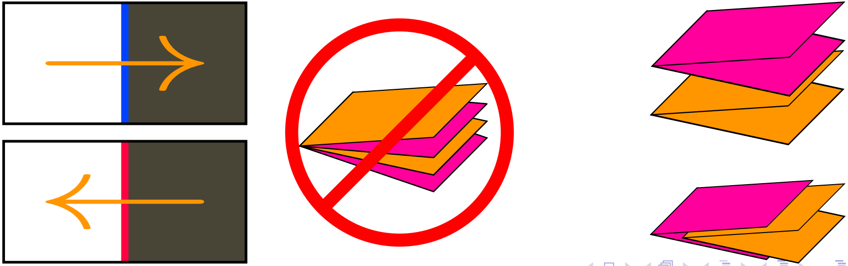


Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

Proof: [$\Longrightarrow$] We have already proven this direction.



Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

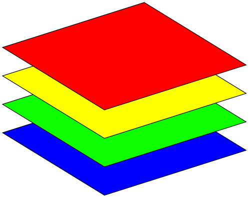
Proof: $[\Longleftarrow]$ Let's prove this direction now.

Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

Proof: $[\Longleftarrow]$ Let's prove this direction now.

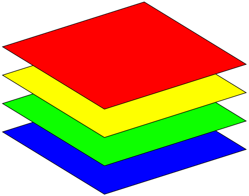


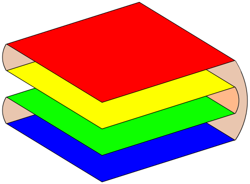
Conditions for flat-foldability

Theorem: (Nishat and Whitesides 2013)

A linear ordering $\mathcal{L}$ of faces is flat-foldable if and only if (i) $\mathcal{L}$ satisfies the partial ordering given by the map and (ii) every pair of twin butterflies stacks or nests in $\mathcal{L}$.

Proof: [$\Longleftarrow$] Let's prove this direction now.





Computational Complexity of Validity Testing

Algorithm for determining whether or not a linear ordering is valid:

Computational Complexity of Validity Testing

Algorithm for determining whether or not a linear ordering is valid:

1. Check that the linear ordering satisfies the partial ordering given by the map.

Computational Complexity of Validity Testing

Algorithm for determining whether or not a linear ordering is valid:

1. Check that the linear ordering satisfies the partial ordering given by the map.
2. Check that every pair of twin butterflies stacks or nests.

Computational Complexity of Validity Testing

Algorithm for determining whether or not a linear ordering is valid:

1. Check that the linear ordering satisfies the partial ordering given by the map.
2. Check that every pair of twin butterflies stacks or nests.

This can be done in $\mathcal{O}(mn)$ (linear) time, which is super fast!

Computational Complexity of Validity Testing

Algorithm for determining whether or not a linear ordering is valid:

1. Check that the linear ordering satisfies the partial ordering given by the map.
2. Check that every pair of twin butterflies stacks or nests.

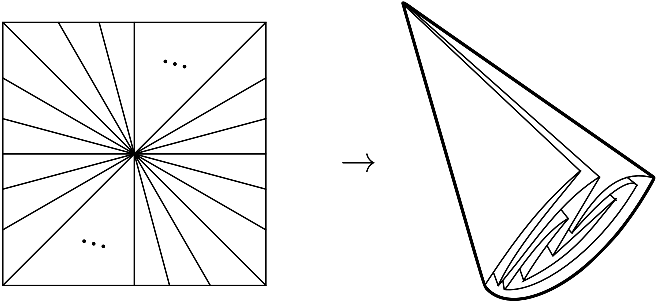
This can be done in $\mathcal{O}(mn)$ (linear) time, which is super fast!

Summary: If someone says "This map can be flat-folded, here is the folding," we can quickly test whether or not they were correct.

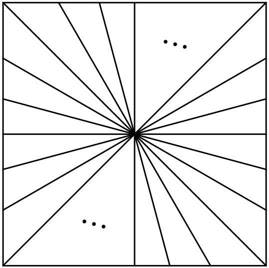
Counting Problems in Origami

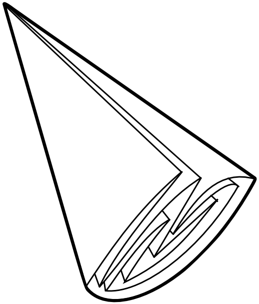
- How many ways can we fold an $n \times m$ map?
  - How many ways can we fold a $1 \times m$ map?
- How many $n \times m$ map crease patterns can be folded, period?
  - How many ways can we fold a $1 \times m$ map?

Star Patterns



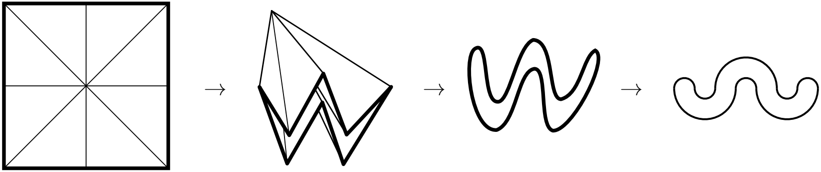
Star Patterns





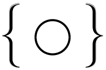
Question: How many foldings are generated by star patterns with $2n$ creases?

Representing foldings of Star Patterns

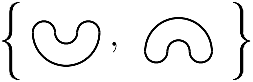


Examples of Meanders

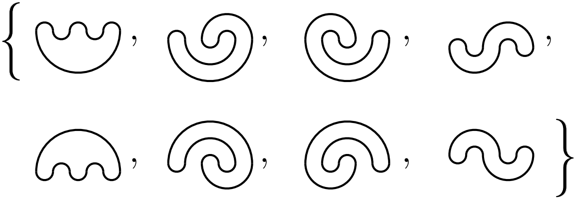
$n = 1$



$n = 2$

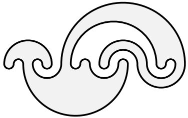


$n = 3$

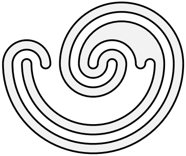


Examples of Meanders

$n = 10$ $\Big\{$ ... ,



,



, ... $\Big\}$

Examples of Meanders

Order n	**# Meanders** M_n
1	1
2	2
3	8
4	42
5	262
$\vdots$	$\vdots$

Examples of Meanders

Order n	**# Meanders** M_n
1	1
2	2
3	8
4	42
5	262
$\vdots$	$\vdots$
10	

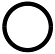
Examples of Meanders

Order n	**# Meanders** M_n
1	1
2	2
3	8
4	42
5	262
⋮	⋮
10	8,152,860
⋮	⋮

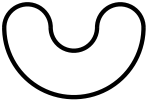
Table: The sequence of Meandric Numbers

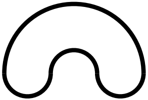
Game Plan

$$n = 1$$

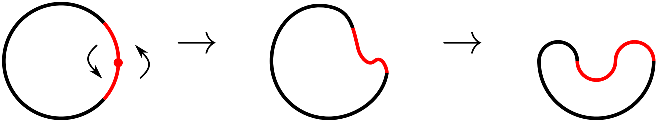


$$n = 2$$

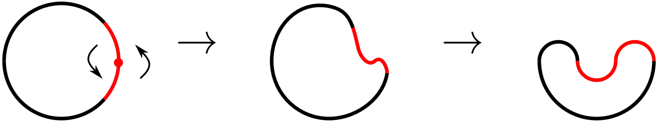




Game Plan

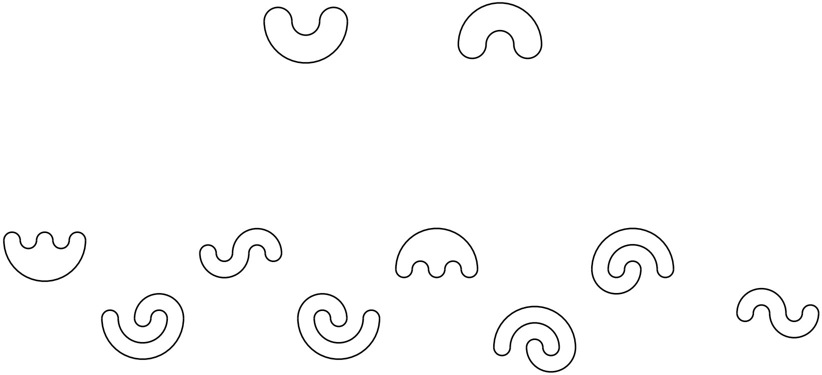


Game Plan

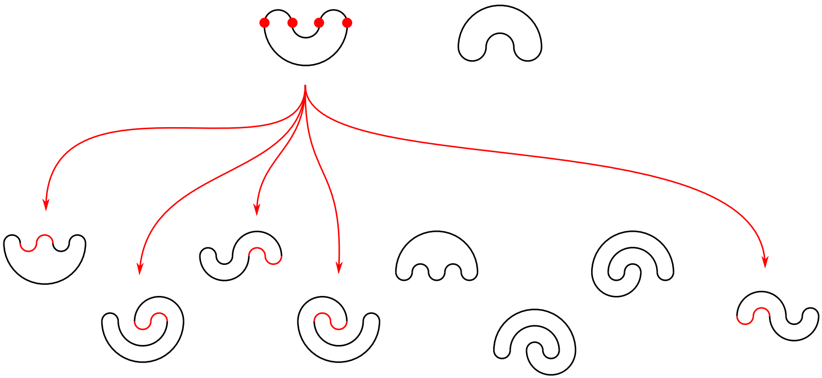


Idea: Produce larger meanders by adding "twists" to smaller meanders

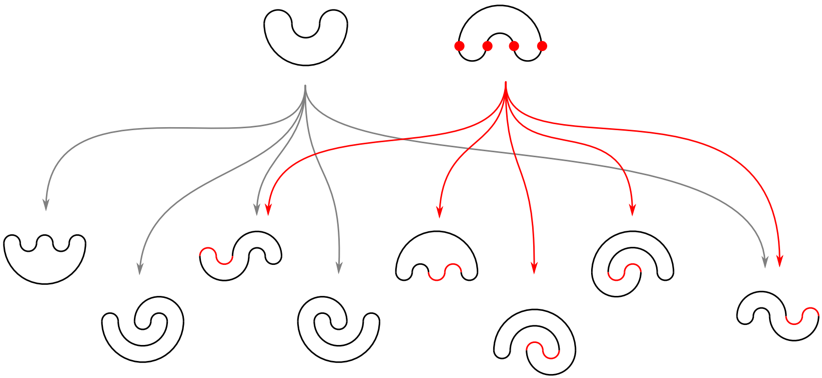
Will it work?



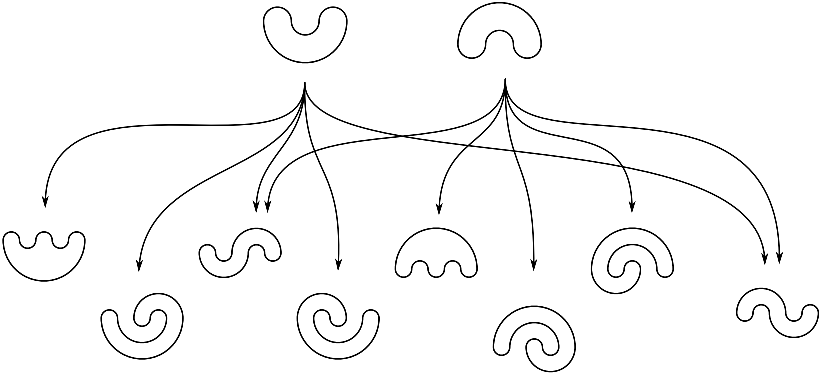
Will it work?



Will it work?



Will it work?

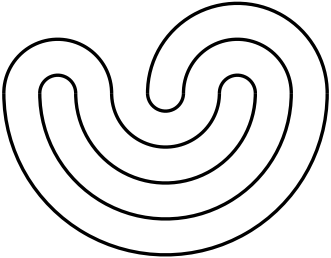


Will it work?

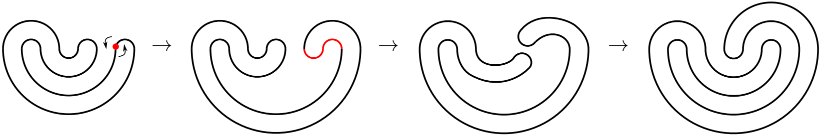
Question: Can we get all meanders by repeatedly doing this?

Will it work?

Question: Can we get all meanders by repeatedly doing this?
Answer: Sadly we cannot:



Will it work?



Meanders

Theorem

All meanders may be built from $m_1 = \circ$ *with at most one shuffle after each twist.*

Simple Meanders

Definition (Simple Meander)

A *simple meander* is a meander of order n that can be constructed without shuffling.

Simple Meanders

Theorem (Recurrence Relation for Simple Meanders)

Let $\mathbb{P}(k,n) = \{(x_1, x_2, \ldots, x_n) : \sum x_i = k \text{ and } x_i \geq 0 \forall i\}$. *Then*

$$r(n) = \sum_{i=1}^{n} \sum_{P \in \mathbb{P}(i,n+1)} \prod_{k=1}^{n+1} r(P_k)$$

$$H(n) = \sum_{i=1}^{n} r(i) \sum_{P \in \mathbb{P}(i,n)} \prod_{k=1}^{n} H(P_k)$$

$$M_n^s = 2H(n)$$

Make sure to stop by our exhibit in the Gould Library!

Opening Spring Term 2016

References

Rahnuma Islam Nishat and Sue Whitesides (2013)
Map Folding
Canadian Conference on Computational Geomery (2013), p49-52