MIT 6.849
Geometric Folding Algorithms
Prof. Erik Demaine

Lecture 6:
Origami Art and Design

Guest Lecturer:
Jason Ku

September 27, 2010

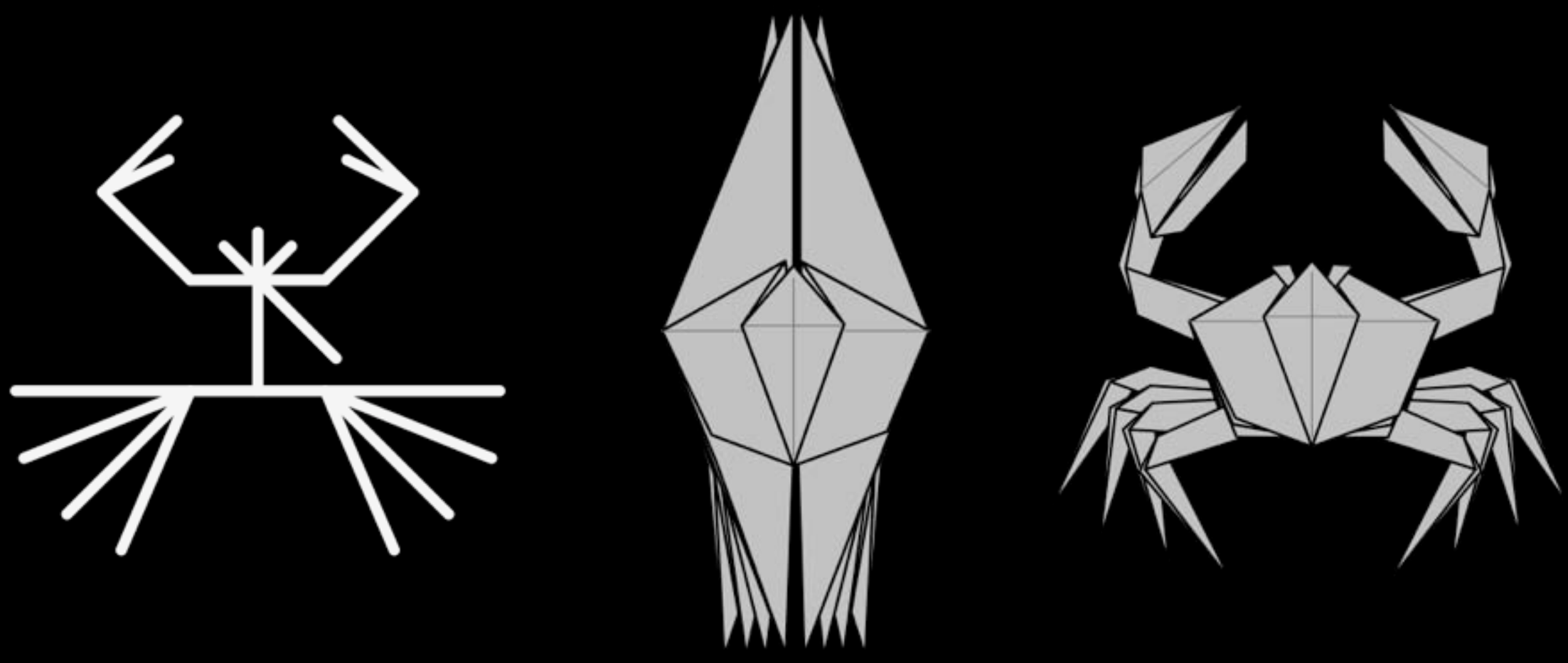
Jason Ku
President of OrigaMIT
Mechanical Engineering Bachelor's, MIT ’09
PhD student in Mechanical Engineering working in folding on the micro and nano scales

Origami Art

- Akira Yoshizawa http://www.origami.vancouver.bc.ca/
- Hideo Komatsu http://www.origami.gr.jp/~komatsu/
- Takashi Hojyo http://origami.gr.jp/~hojyo
- David Brill http://www.brilliantorigami.com/
- Michael LaFosse http://www.origamido.com
- Eric Joisel http://www.ericjoisel.com
- Robert Lang http://www.langorigami.com/
- Brian Chan http://chosetec.darkclan.net/origami/
- Satoshi Kamiya http://www.folders.jp/
- Jason Ku http://scripts.mit.edu/~jasonku/

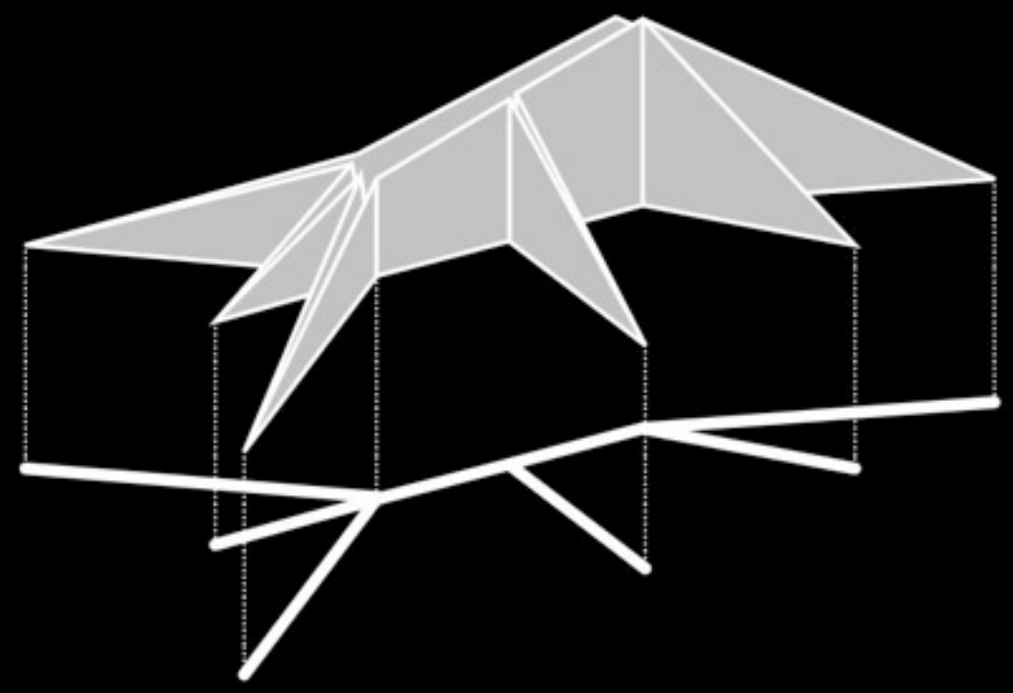
Websites where photos in presentation come from
Origami Art comparison with Music
Representational origami = traditionally represent living things

Tree Theory Review



Review of Tree Theory thought process
1) Start with object
2) Draw tree
3) Change tree into uniaxial base
4) Shape uniaxial base

Uniaxial Bases



1. in z≥0 half plane

2. intersection with z=0 plane = projection onto the plane

3. partition of faces into flaps, each projecting to a line segment

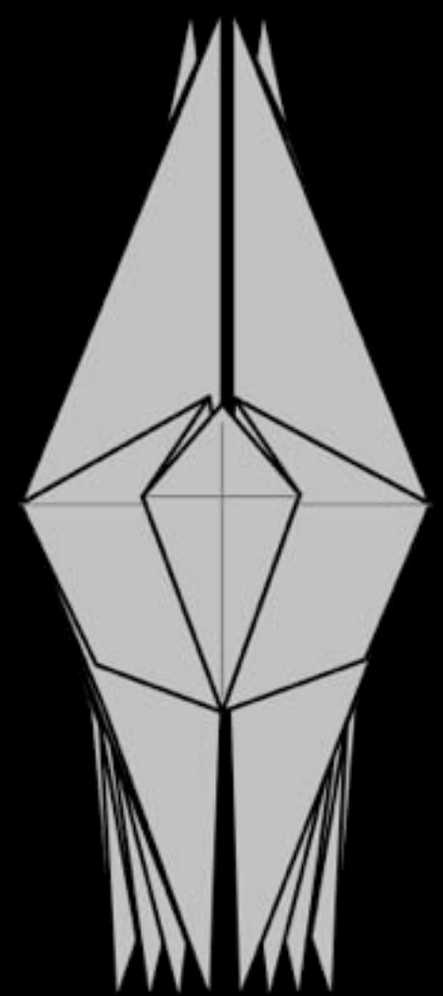
4. hinge crease shared by two flaps project to a point

5. graph of flap projections as edges is a tree

6. only one point of paper folds to each leaf

Previous definition of uniaxial bases
(6) not necessary but convenient
Why would it be useful to have the end of a leaf node map to more than one point on paper? Ans: flap thickness at end
What does this really mean?

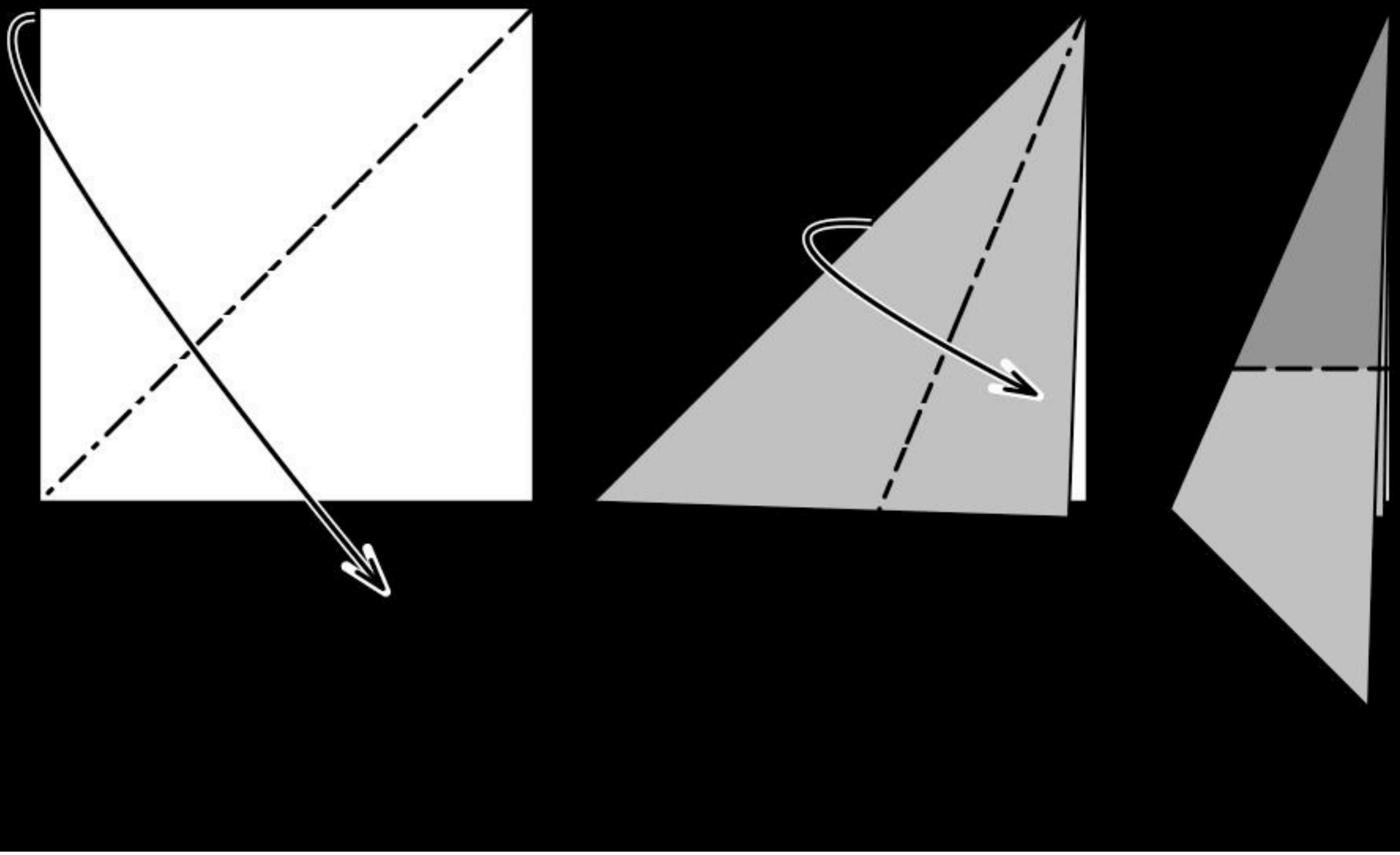
Uniaxial Bases



1. flaps lie along or straddle a single line (the axis)
2. flaps hinge perpendicular to the axis
3. can thin to stick figure (tree)

Informal definition are bases that can be trivially modified to become unaxial

Flaps

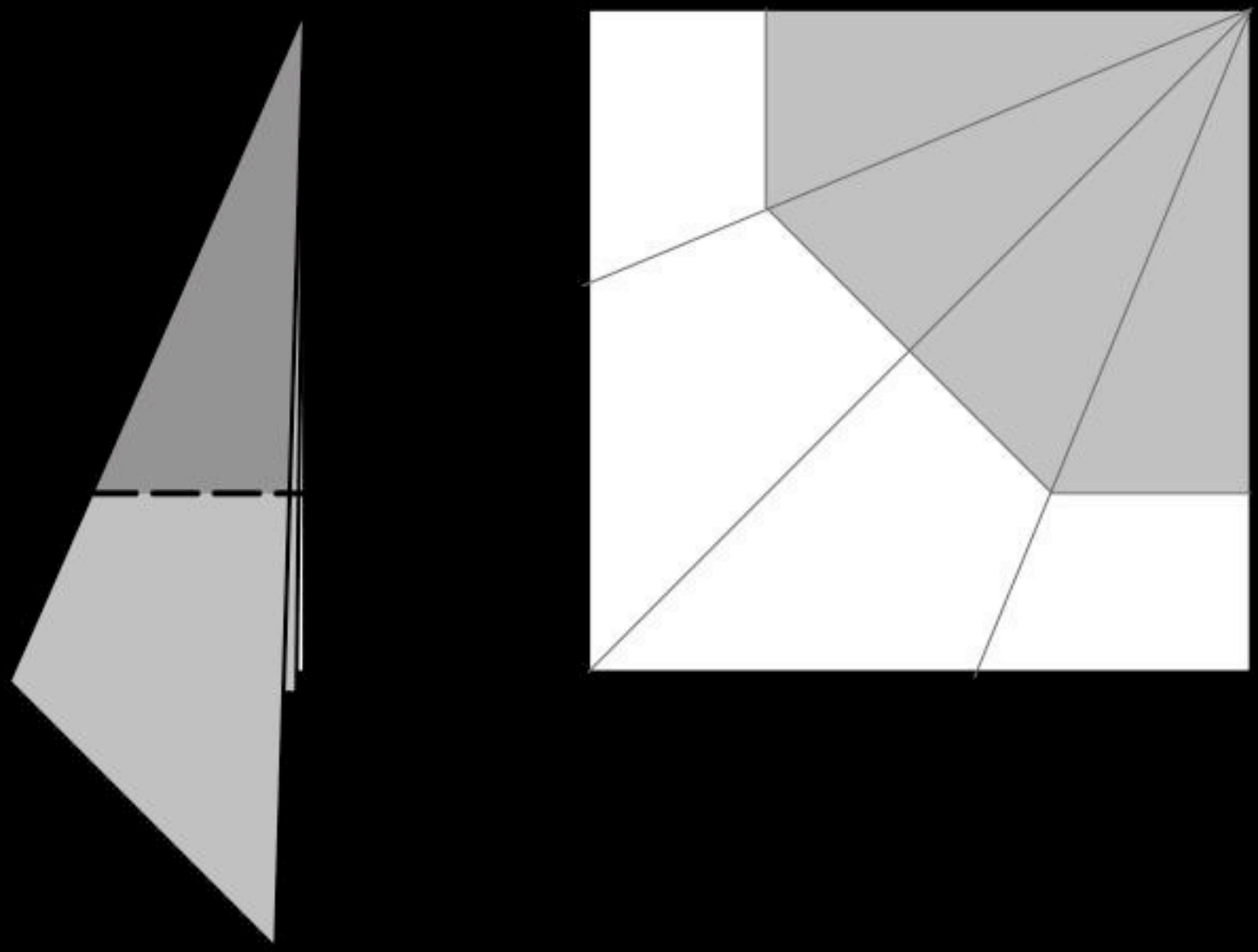


6

Modeling a flap

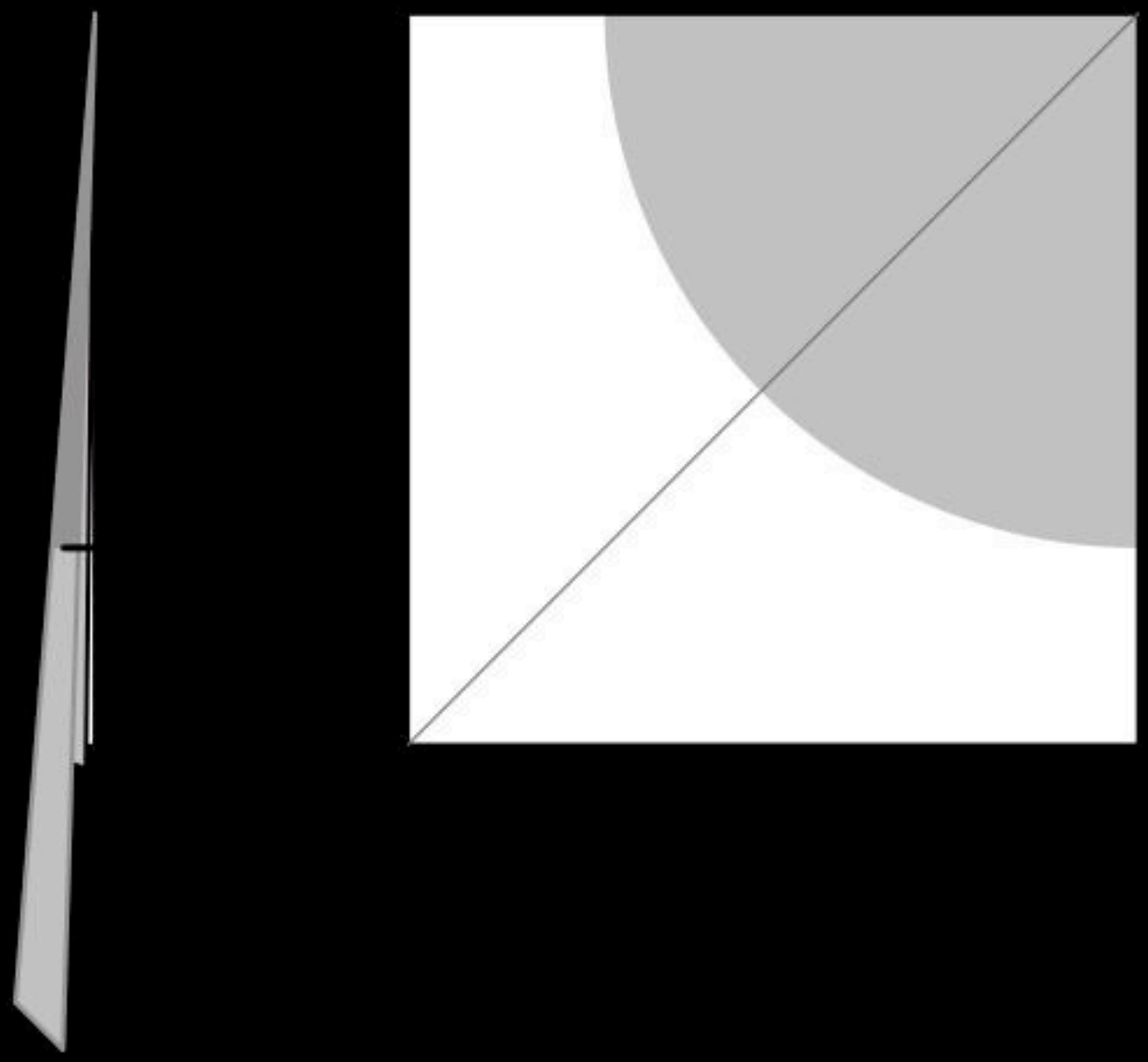
Courtesy of Jason Ku. Used with permission.

Flaps



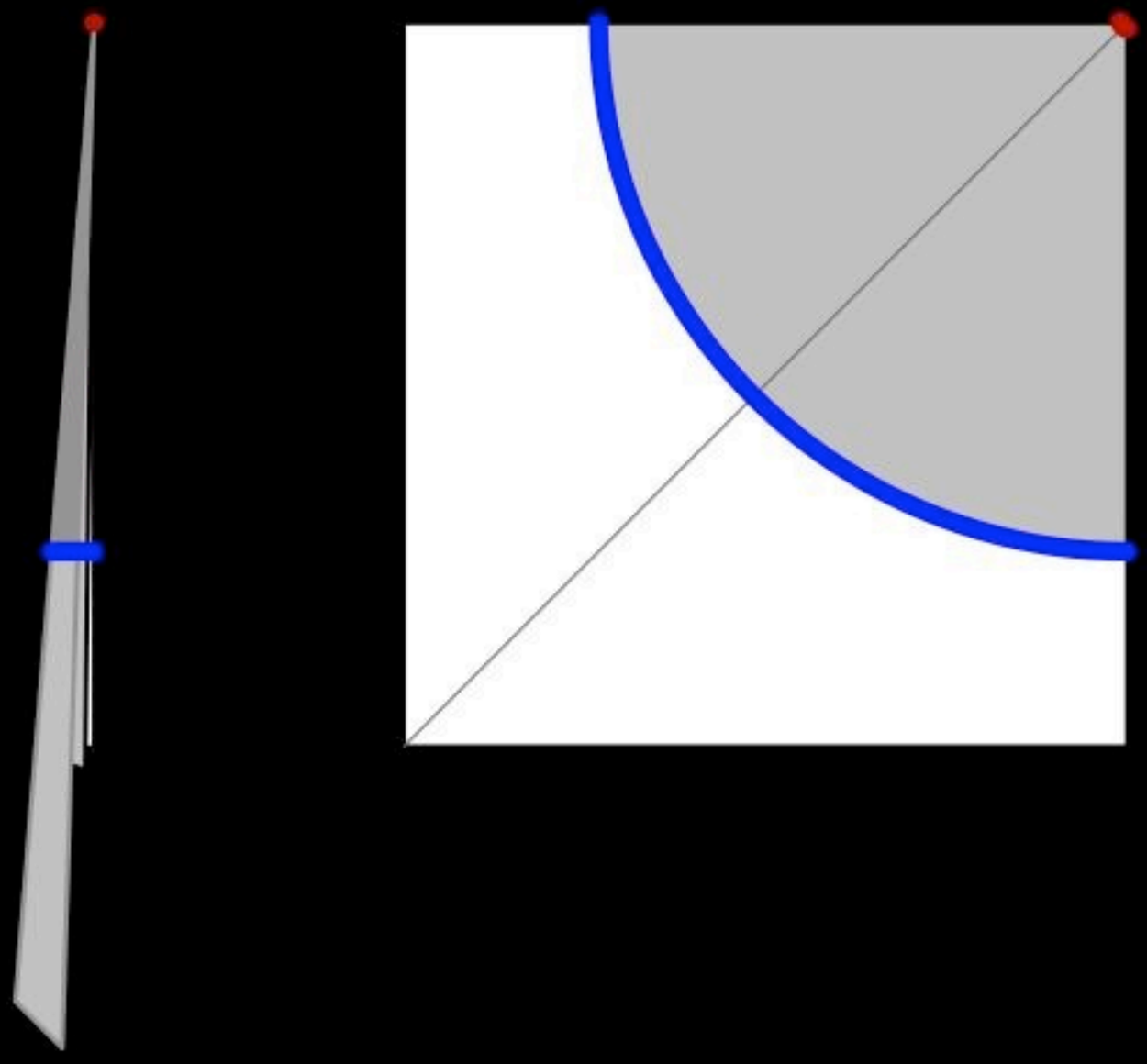
Idea of ‘elevation’ on a flap/tree edge
Rivers separate two parts of a tree with strip of constant width
Circle limiting case of river separating single point from rest
Splitting a leaf edge into a leaf and brach creates a redundant node

Flaps



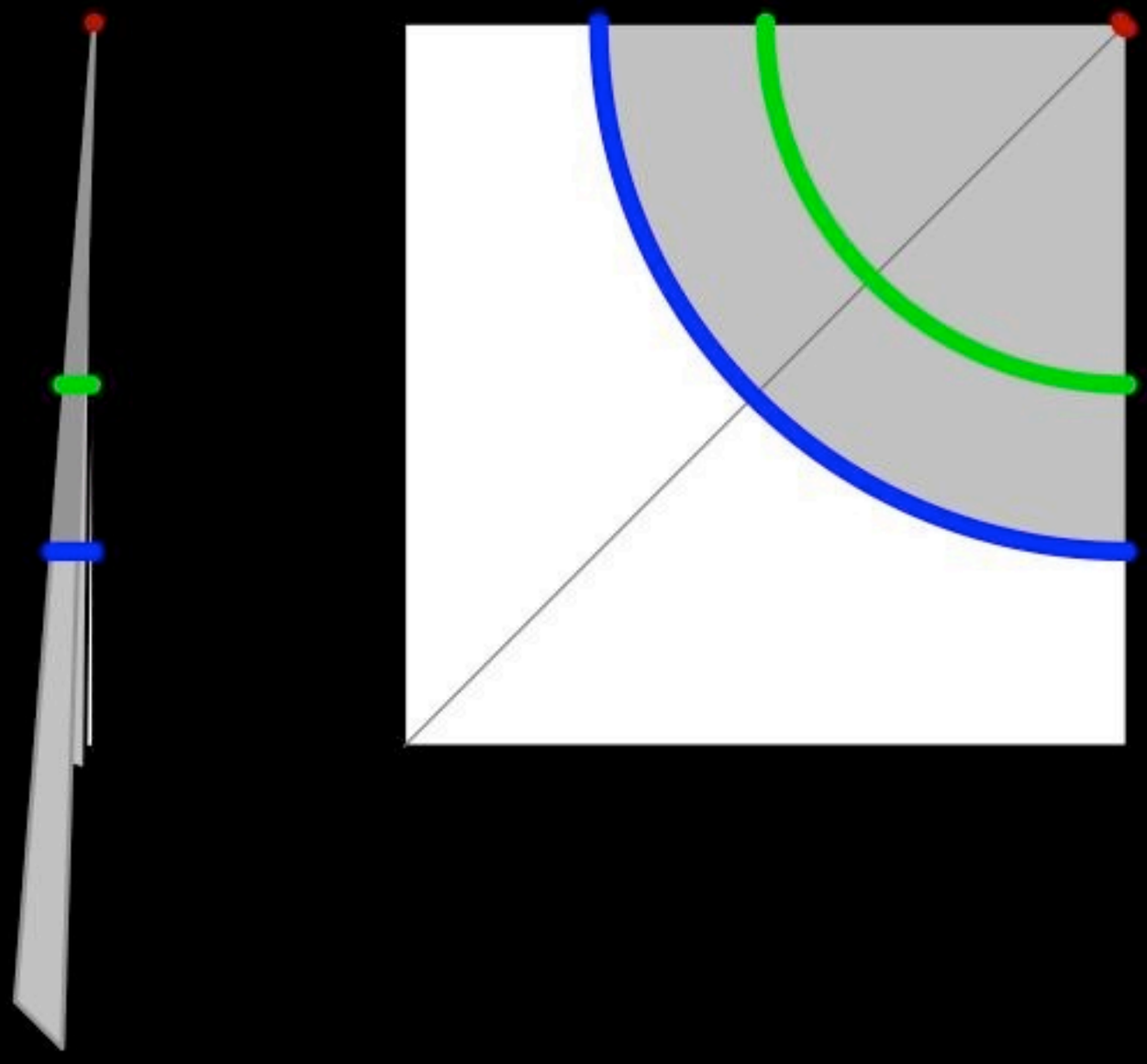
Idea of ‘elevation’ on a flap/tree edge
Rivers separate two parts of a tree with strip of constant width
Circle limiting case of river separating single point from rest
Splitting a leaf edge into a leaf and brach creates a redundant node

Flaps



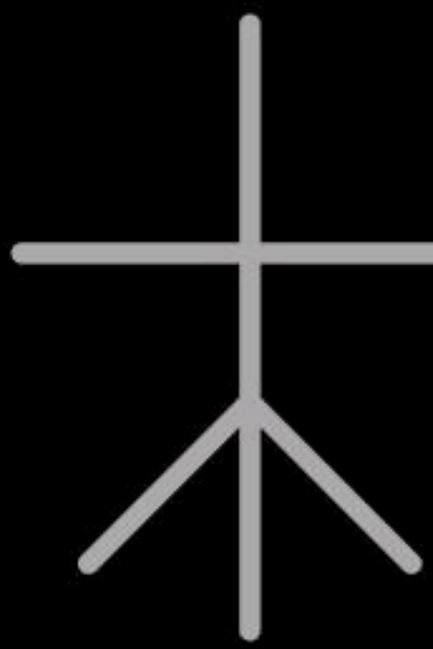
Idea of ‘elevation’ on a flap/tree edge
Rivers separate two parts of a tree with strip of constant width
Circle limiting case of river separating single point from rest
Splitting a leaf edge into a leaf and brach creates a redundant node

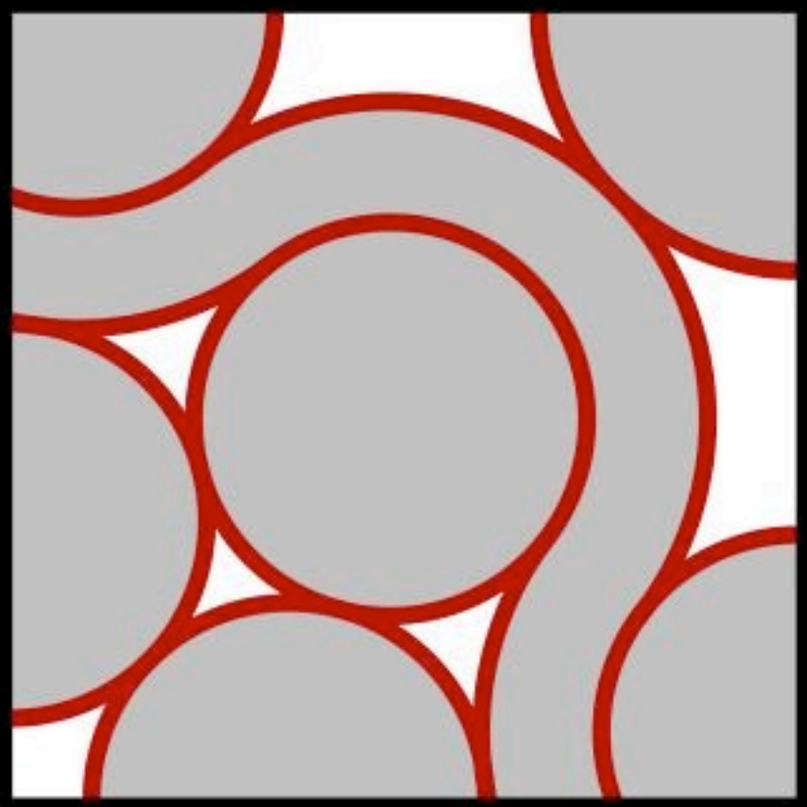
Flaps



Idea of ‘elevation’ on a flap/tree edge
Rivers separate two parts of a tree with strip of constant width
Circle limiting case of river separating single point from rest
Splitting a leaf edge into a leaf and brach creates a redundant node

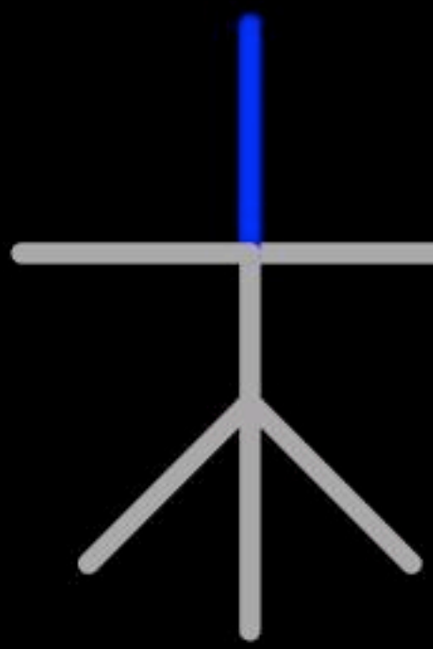
Flaps

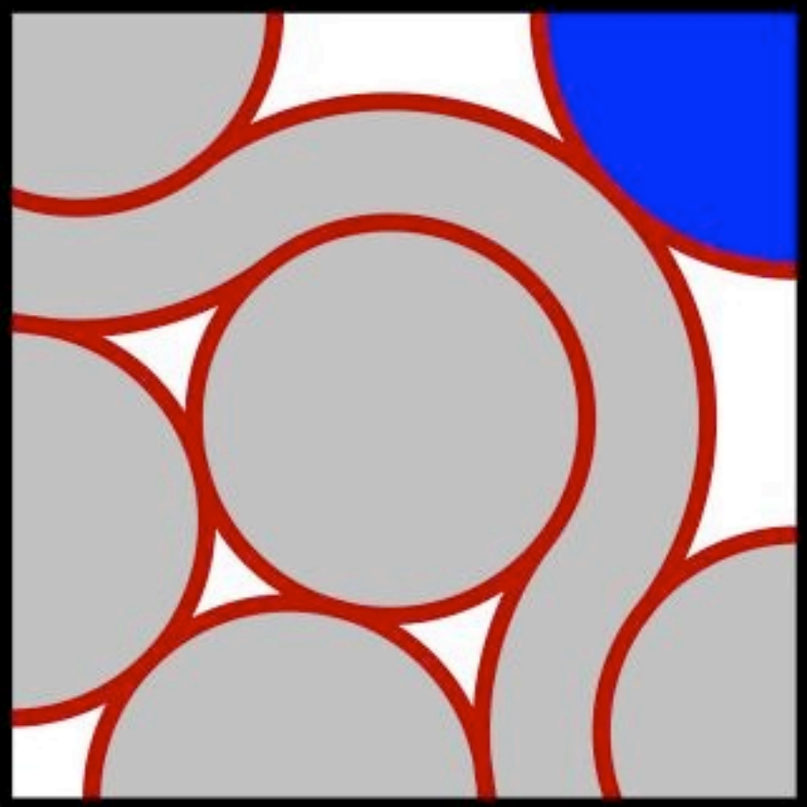




Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

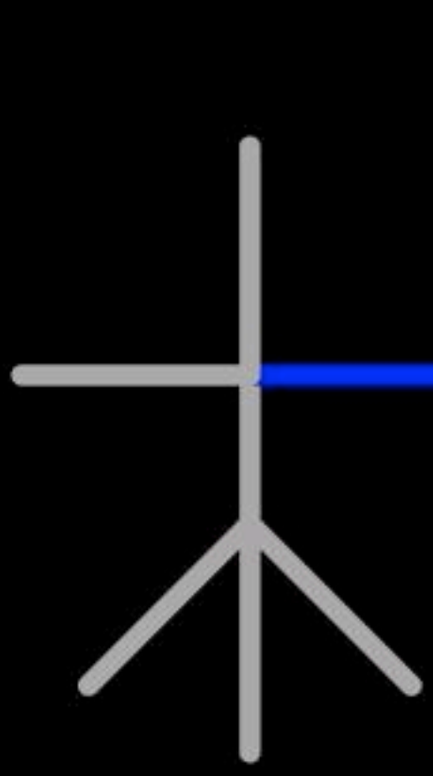
Flaps

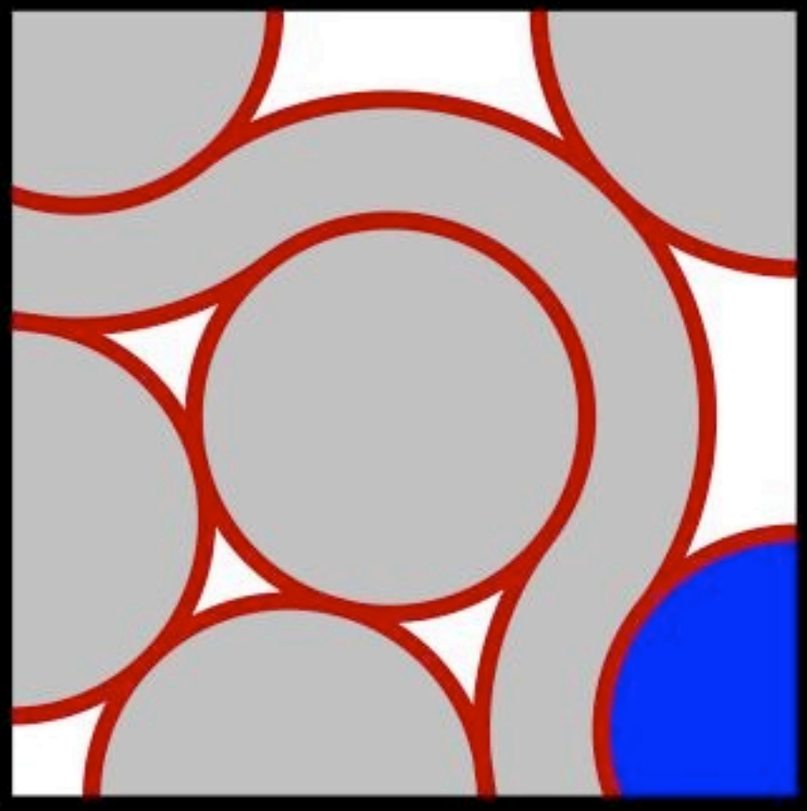




Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

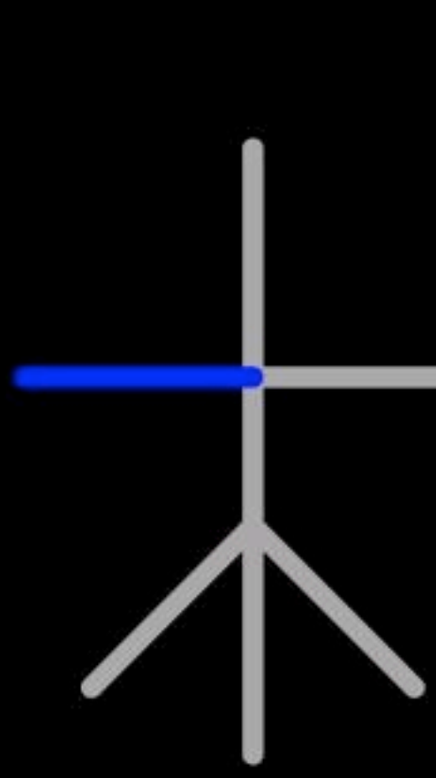
Flaps

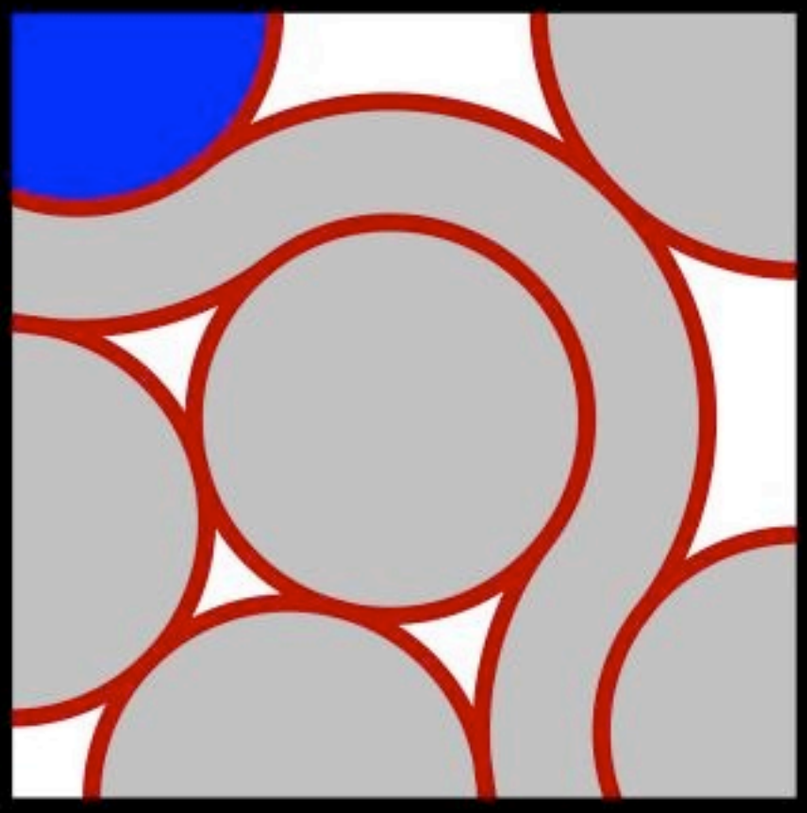




Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

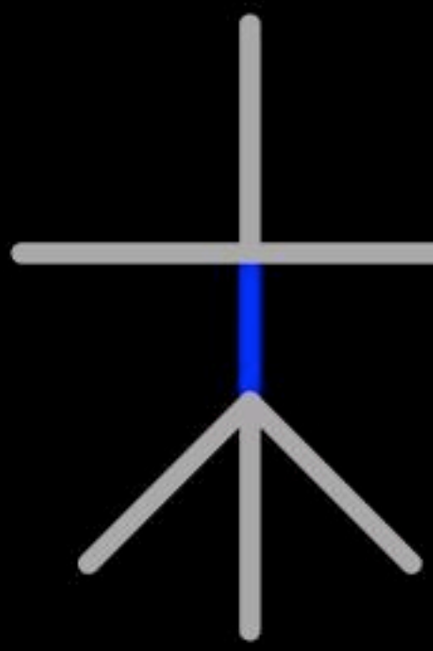
Flaps

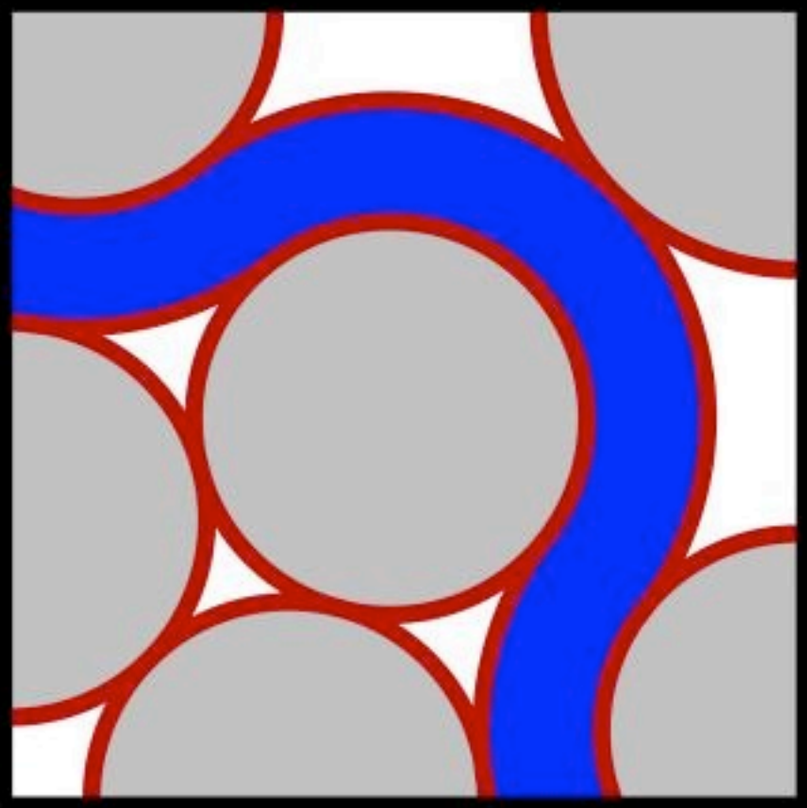




Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

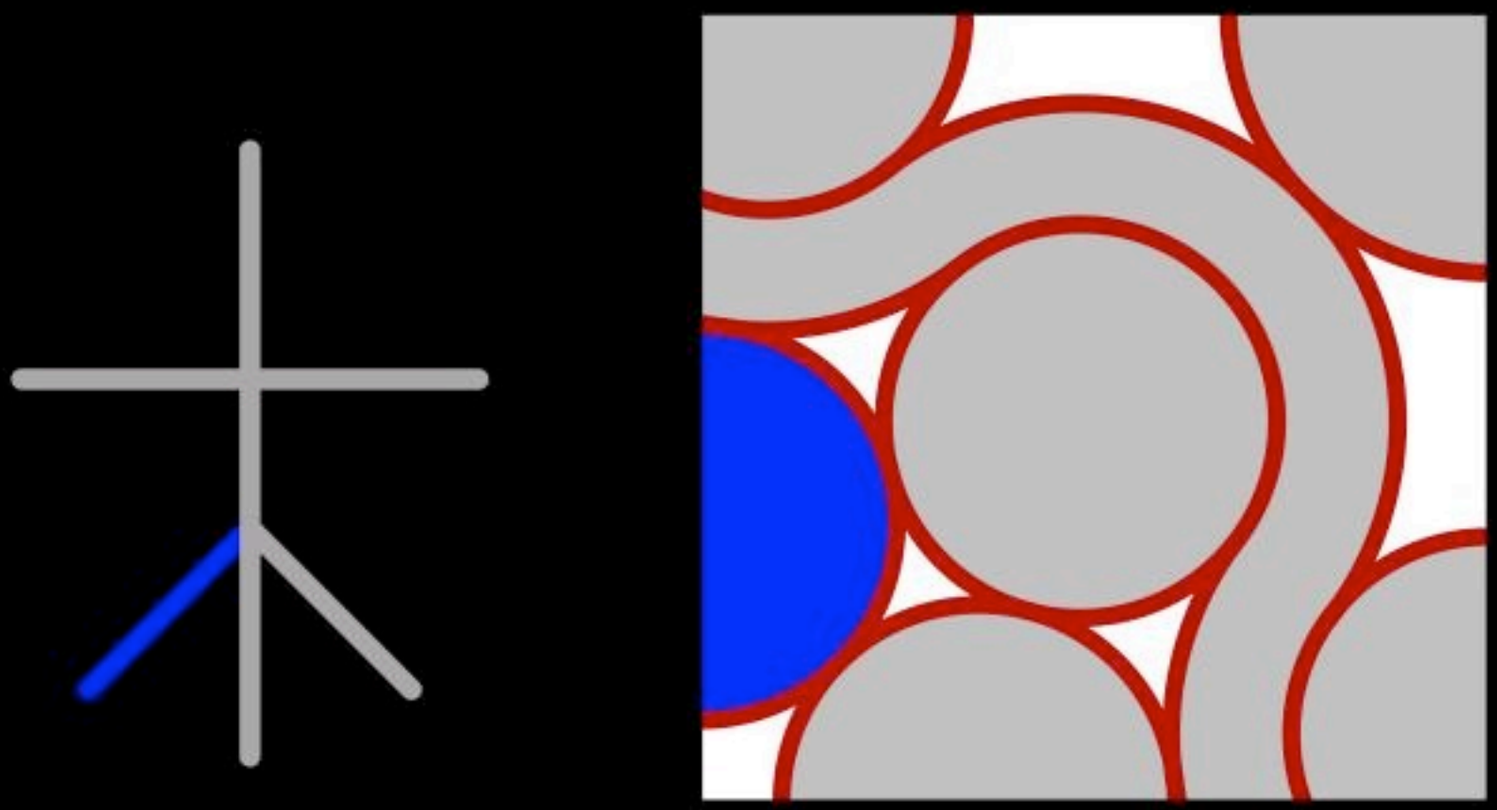
Flaps





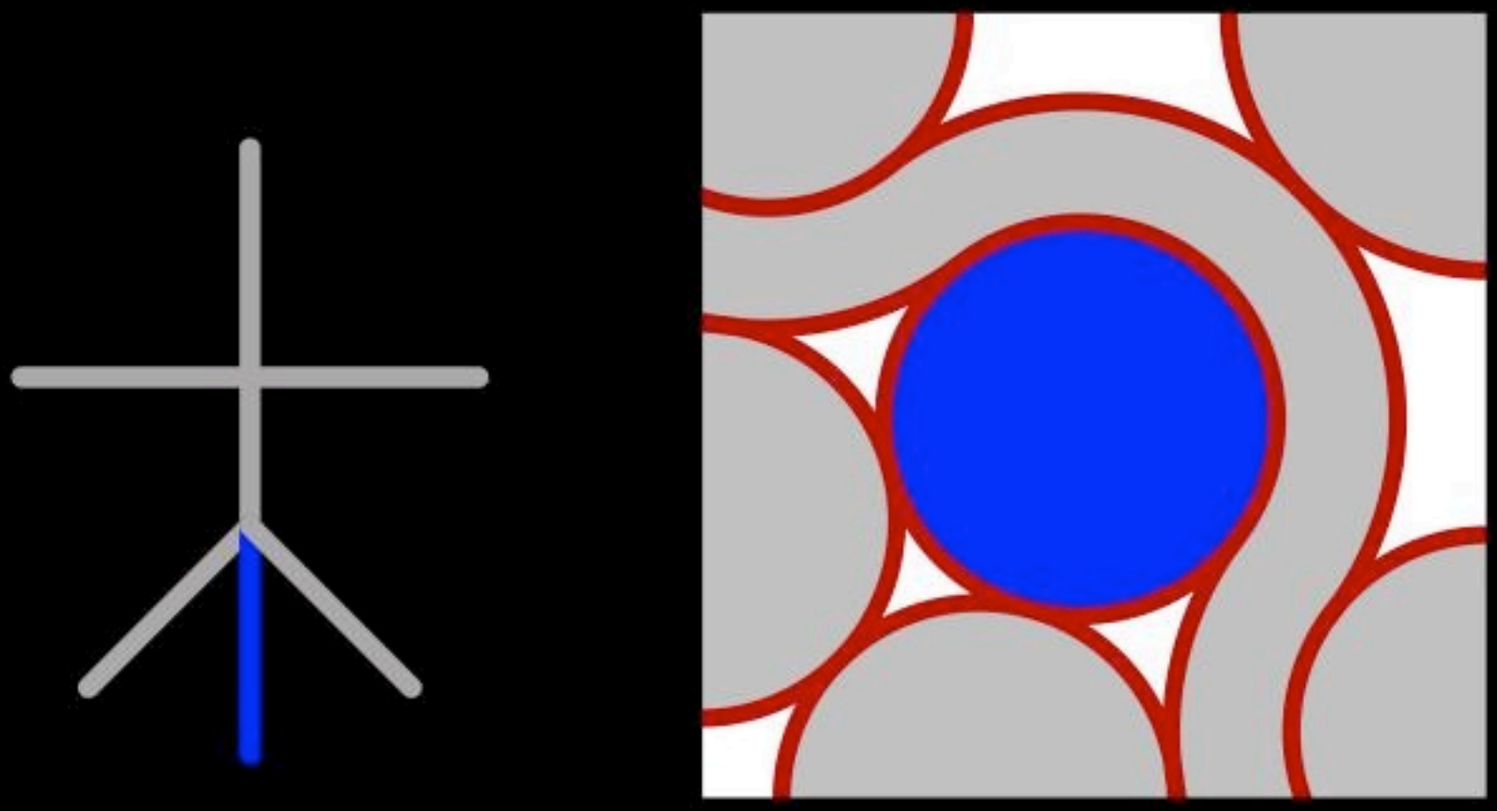
Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

Flaps



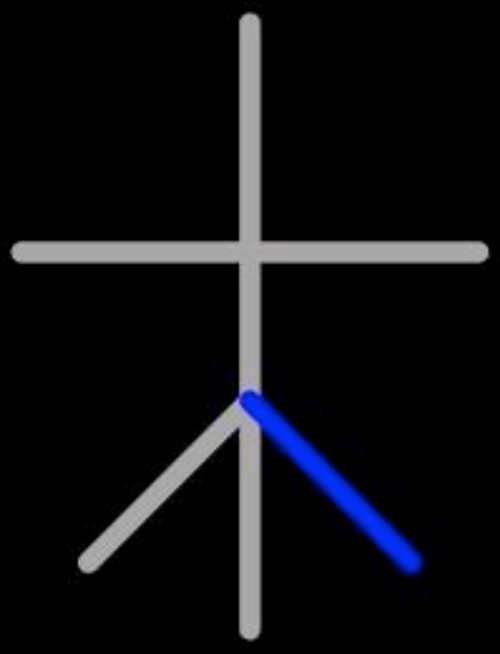
Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

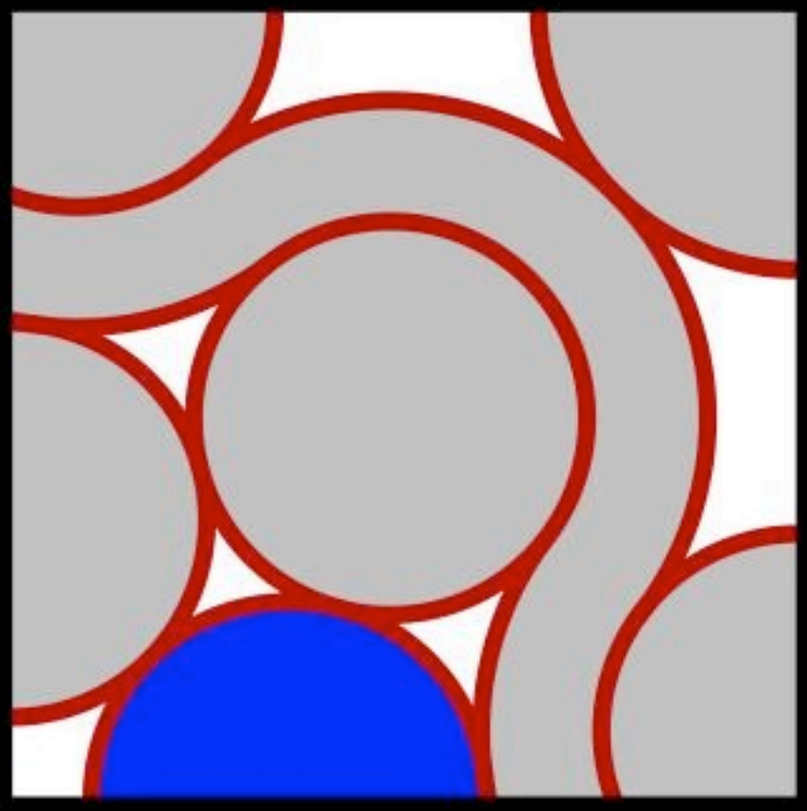
Flaps



Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

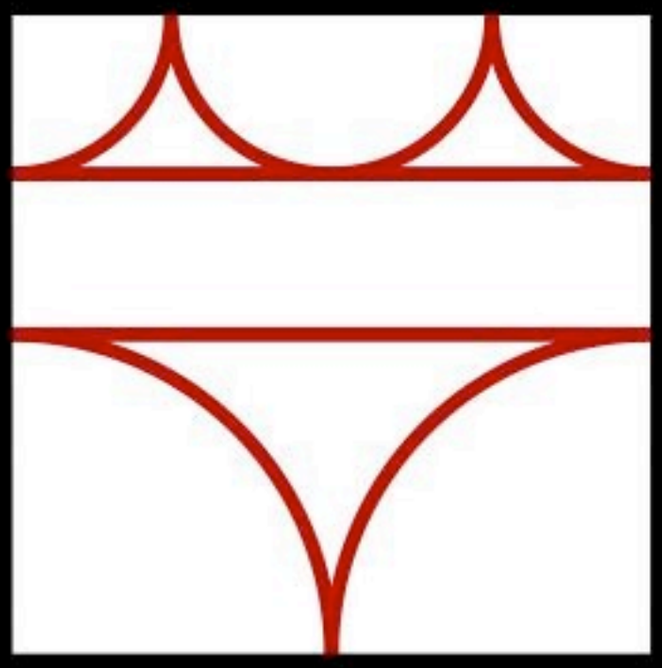
Flaps





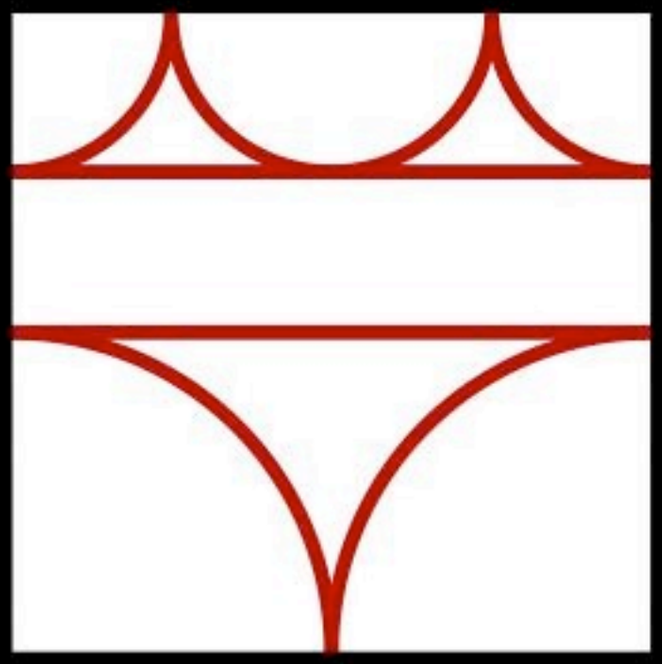
Circle/River Packing (CRP) as a space allocation
Uniquely defines a tree
Tree edges can be oriented anyway we like because if uniaxial base is infinitely thinned, base is actually stick figure
Space between circles is wasted paper and maps to a single tree node

Practice!



Which trees represent the given CRP?

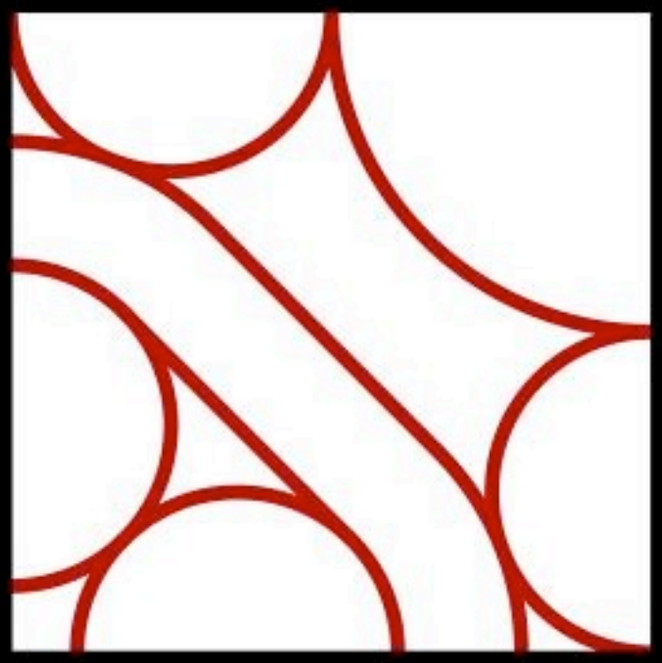
Practice!



20

Which trees represent the given CRP?

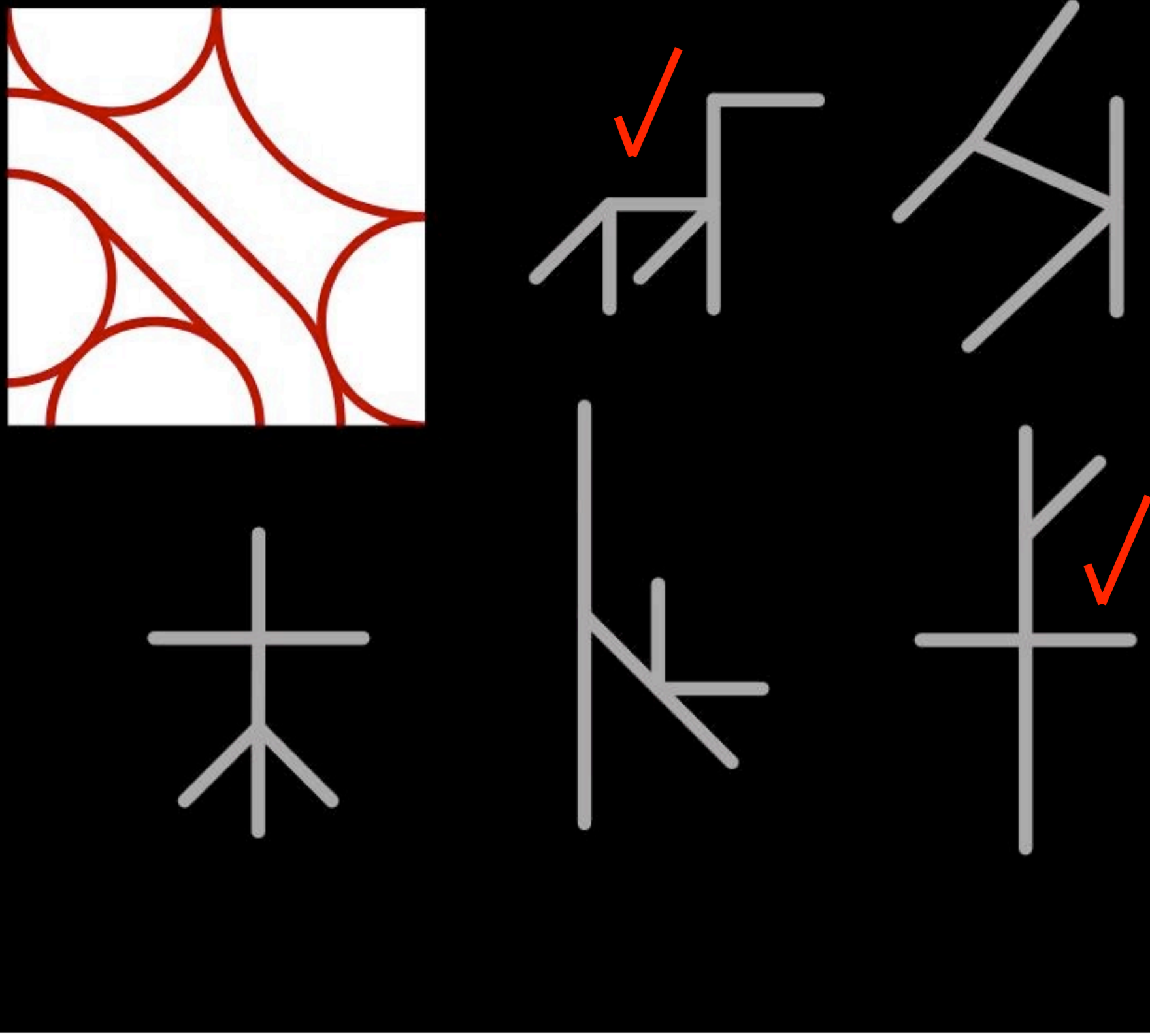
Practice!



21

Which trees represent the given CRP?

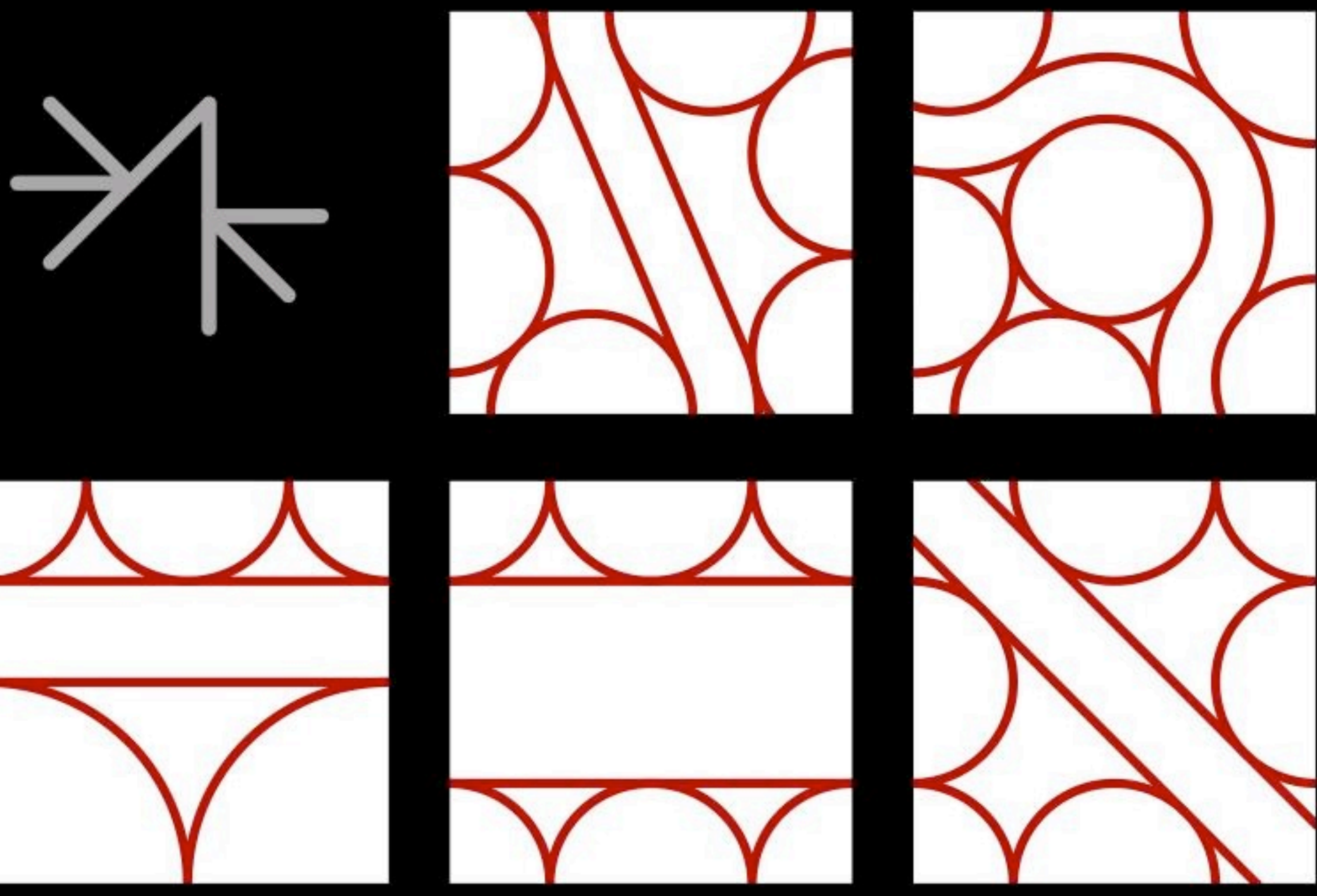
Practice!



22

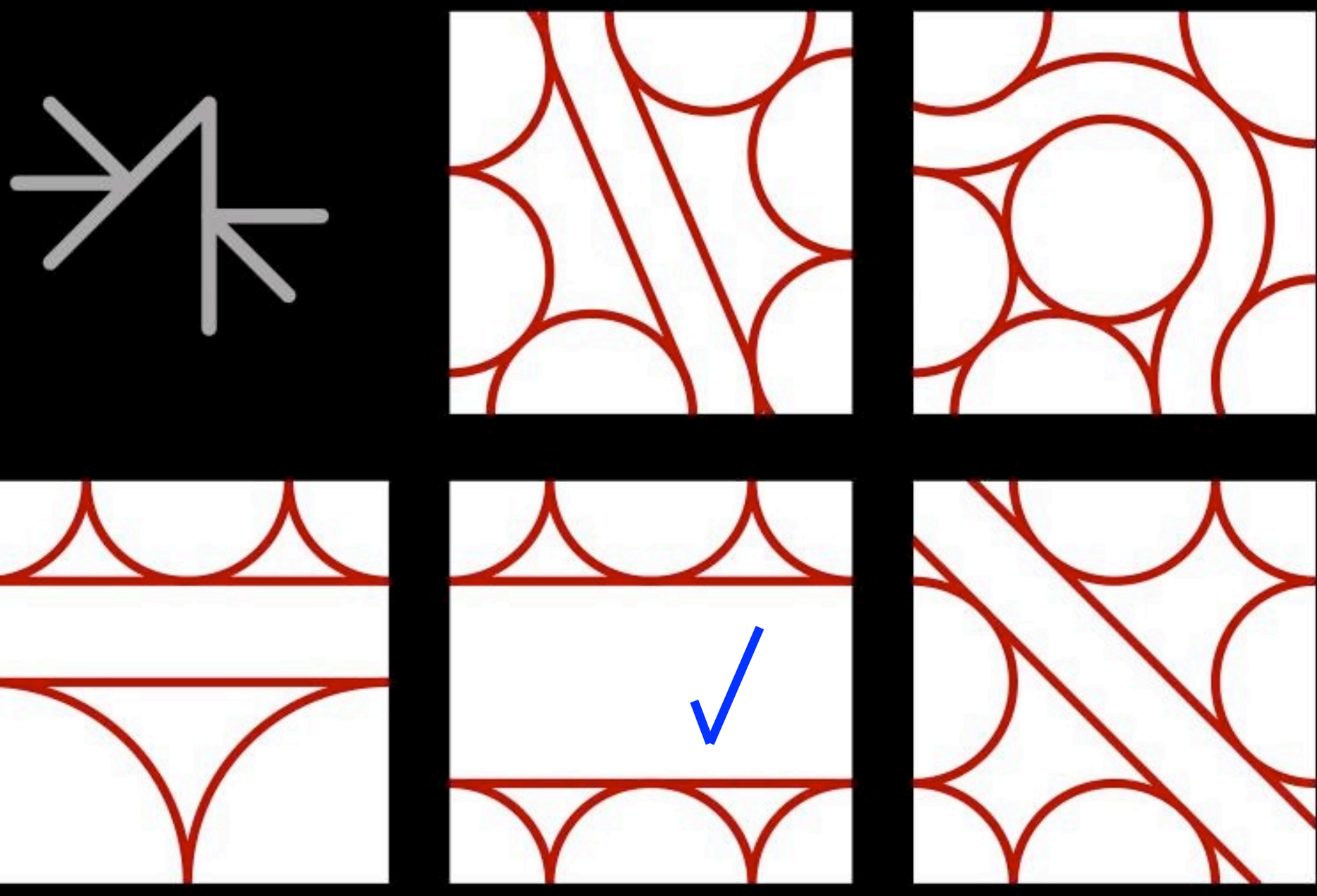
Which trees represent the given CRP?

Practice!



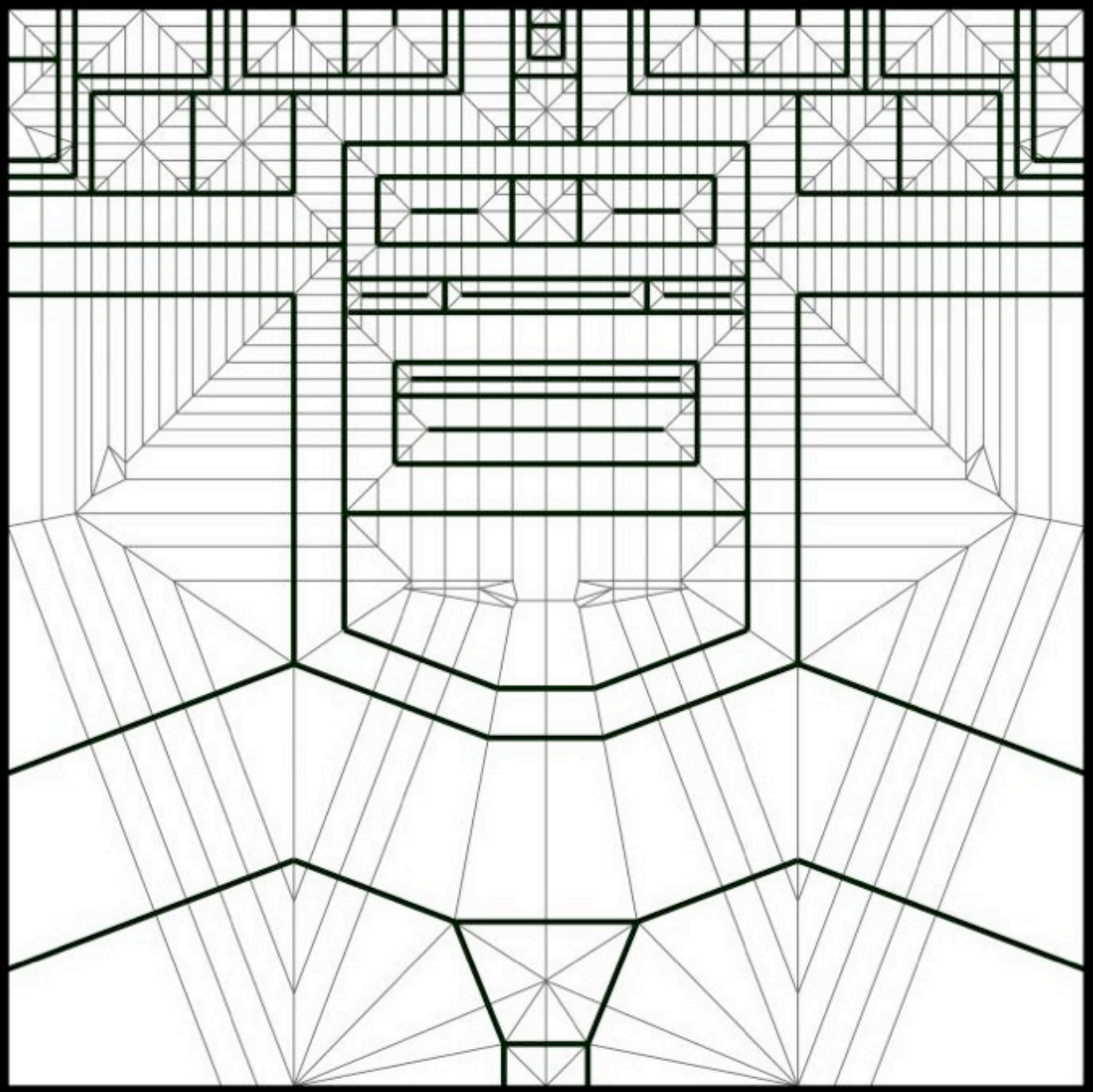
Which CRP correspond to the given tree?
CRP 1, 2, and 5 have similar trees, but different space allocation
(CRP => Tree) = unique
(Tree => CRP) = non-unique

Practice!



Which CRP correspond to the given tree?
CRP 1, 2, and 5 have similar trees, but different space allocation
(CRP => Tree) = unique
(Tree => CRP) = non-unique

Model vs. Reality

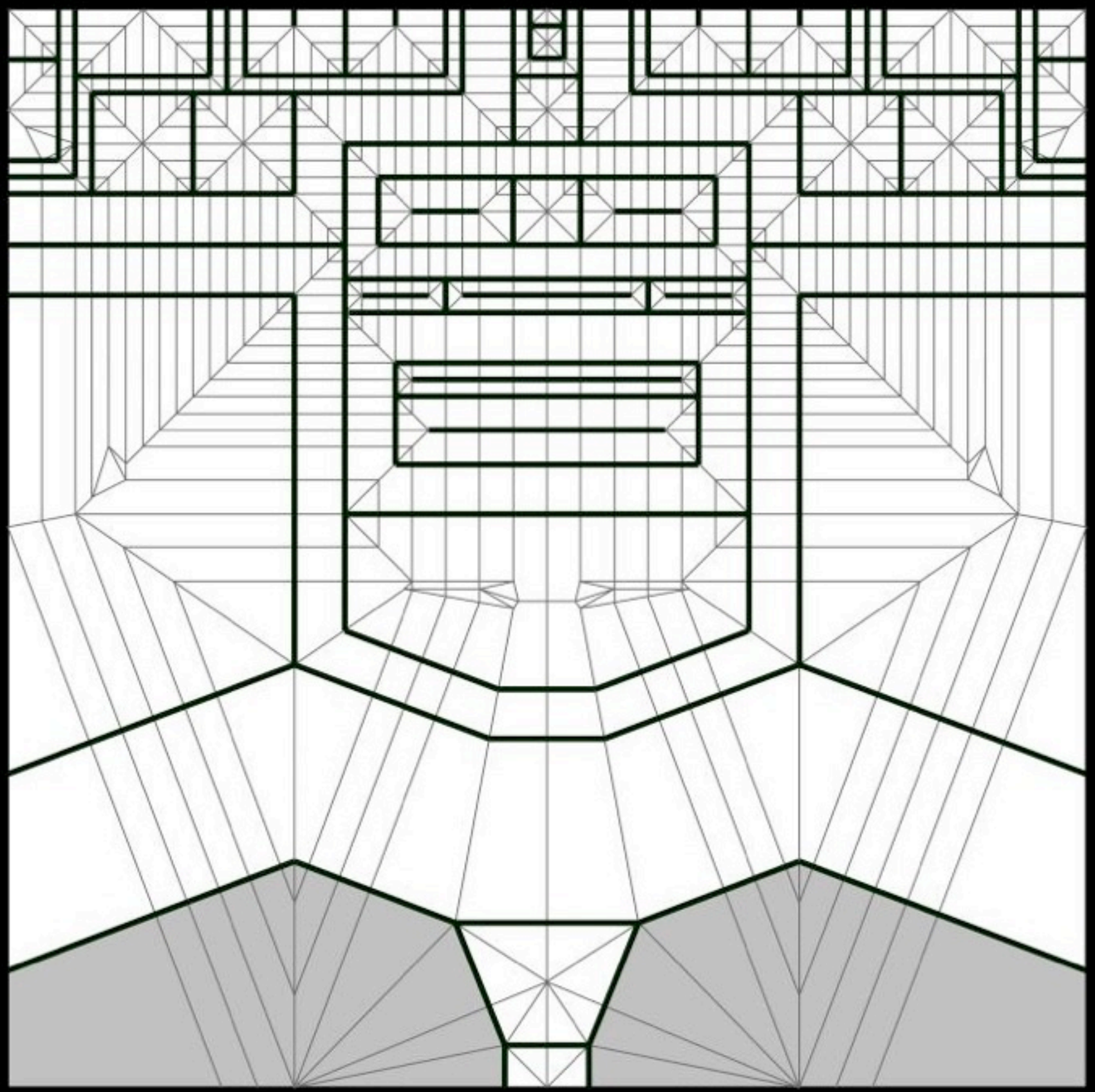


In reality, CRP is an idealization

By definition, locus of all possible hinge creases represents something topologically similar to a CRP

Can read off tree as before

Model vs. Reality

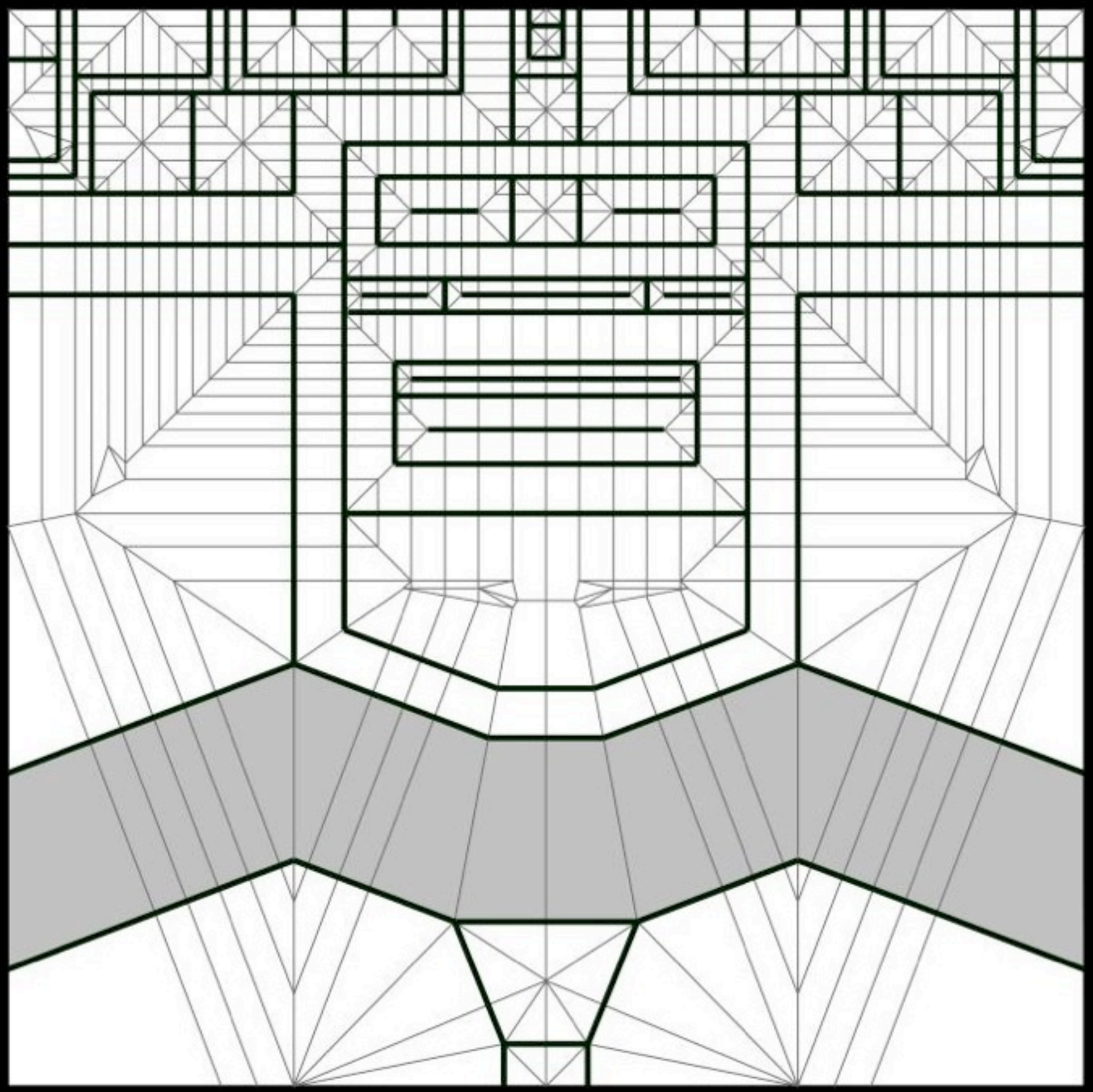


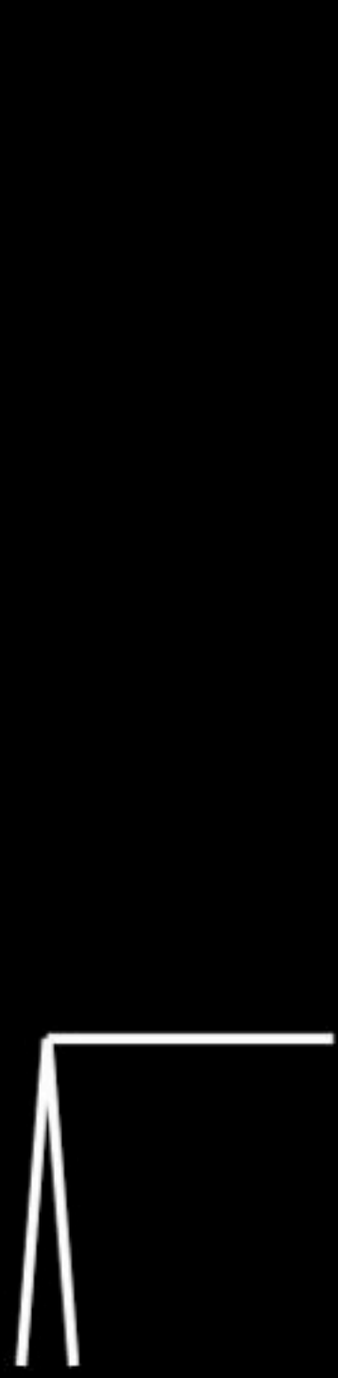
In reality, CRP is an idealization

By definition, locus of all possible hinge creases represents something topologically similar to a CRP

Can read off tree as before

Model vs. Reality



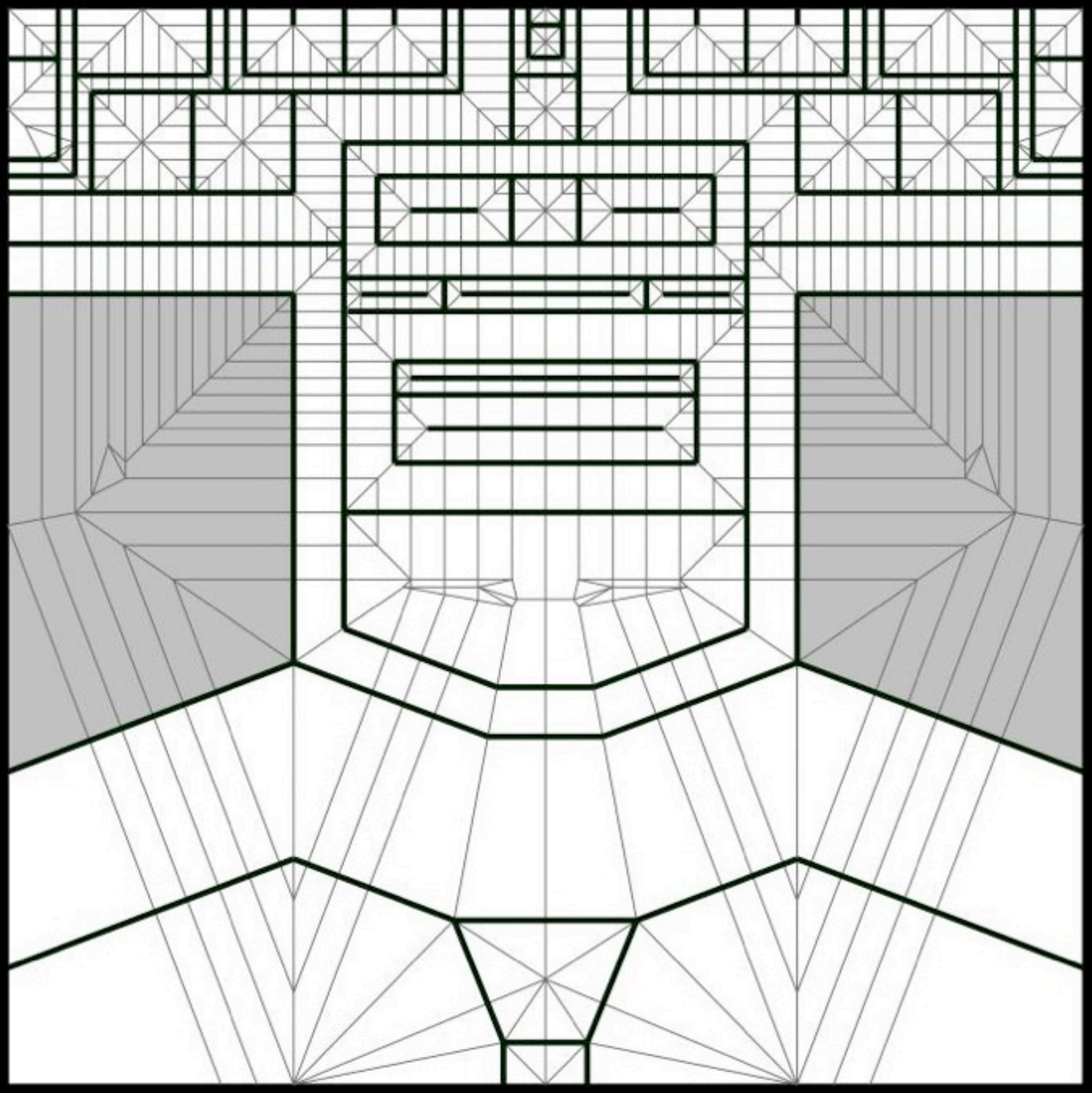


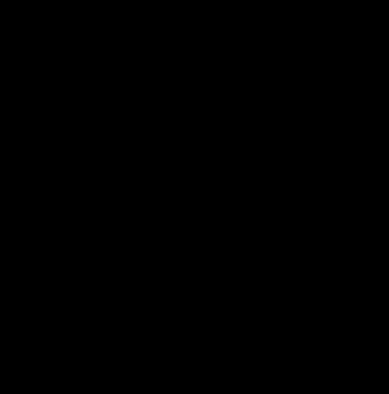
In reality, CRP is an idealization

By definition, locus of all possible hinge creases represents something topologically similar to a CRP

Can read off tree as before

Model vs. Reality



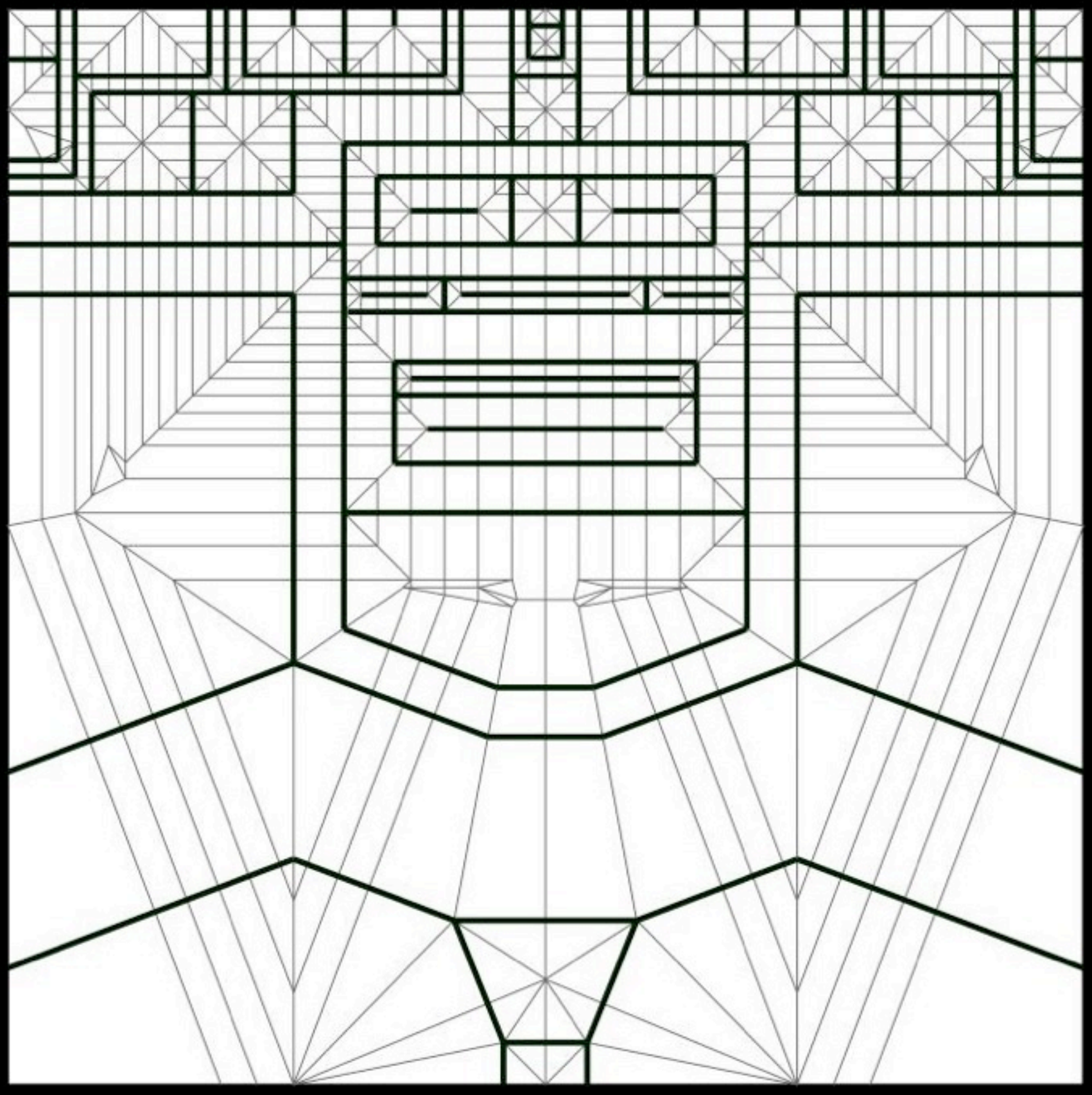


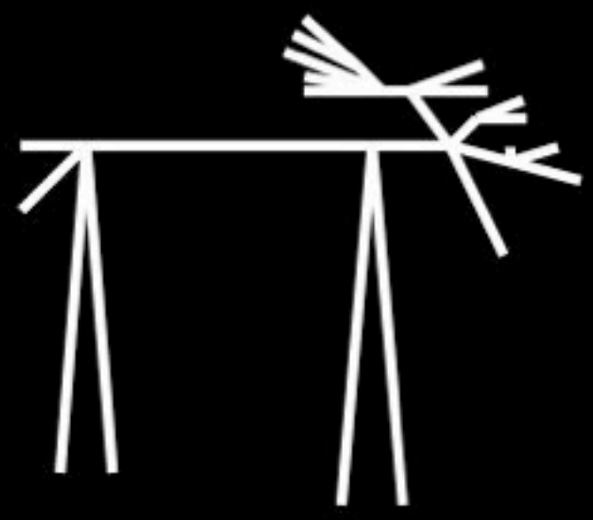
In reality, CRP is an idealization

By definition, locus of all possible hinge creases represents something topologically similar to a CRP

Can read off tree as before

Model vs. Reality





In reality, CRP is an idealization

By definition, locus of all possible hinge creases represents something topologically similar to a CRP

Can read off tree as before

Jason Ku – 2008

Courtesy of Jason Ku. Used with permission.

Design Example

Modeling a crab
First draw tree (blackboard)

Courtesy of Jason Ku. Used with permission.

TreeMaker Example

Symmetry (book/diagonal)
Identifying/fixing unconstrained nodes with local strain
Triangulation of creasepattern (need three degrees of freedom)
View Settings

Useful Features in TreeMaker

Conditions

- axis of symmetry conditions
- force paths to be active or at specific angles
- force nodes to edge/corner/specific locations

Tree manipulation

- adding local strain (Menu/Action/Scale Selection/)
- triangulation (Menu/Edit/Stub/Triangulate Tree/)

Views

- Menu/View/Show View Settings/ very useful
- Can view just locus of hinge creases by turning off all but (Creases/Minor Creases) and (Creases/Lines)

Symmetry (book/diagonal)
Identifying/fixing unconstrained nodes with local strain
Triangulation of creasepattern (need three degrees of freedom)
View Settings

Possible Problems in Optimization

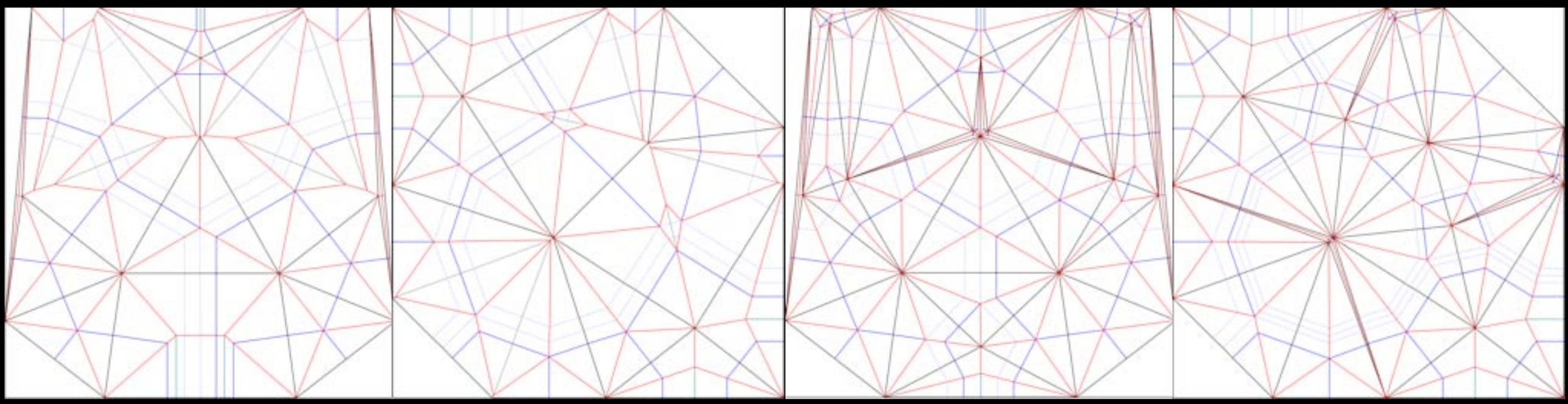
Problem: A polygon bounded by active paths is concave
Solution: add extra leaf node in interior & expand (split polygon into multiple convex polygons)

Problem: A polygon bounded by active paths contains an unconstrained node
Solution: add local strain to interior node to create additional active paths

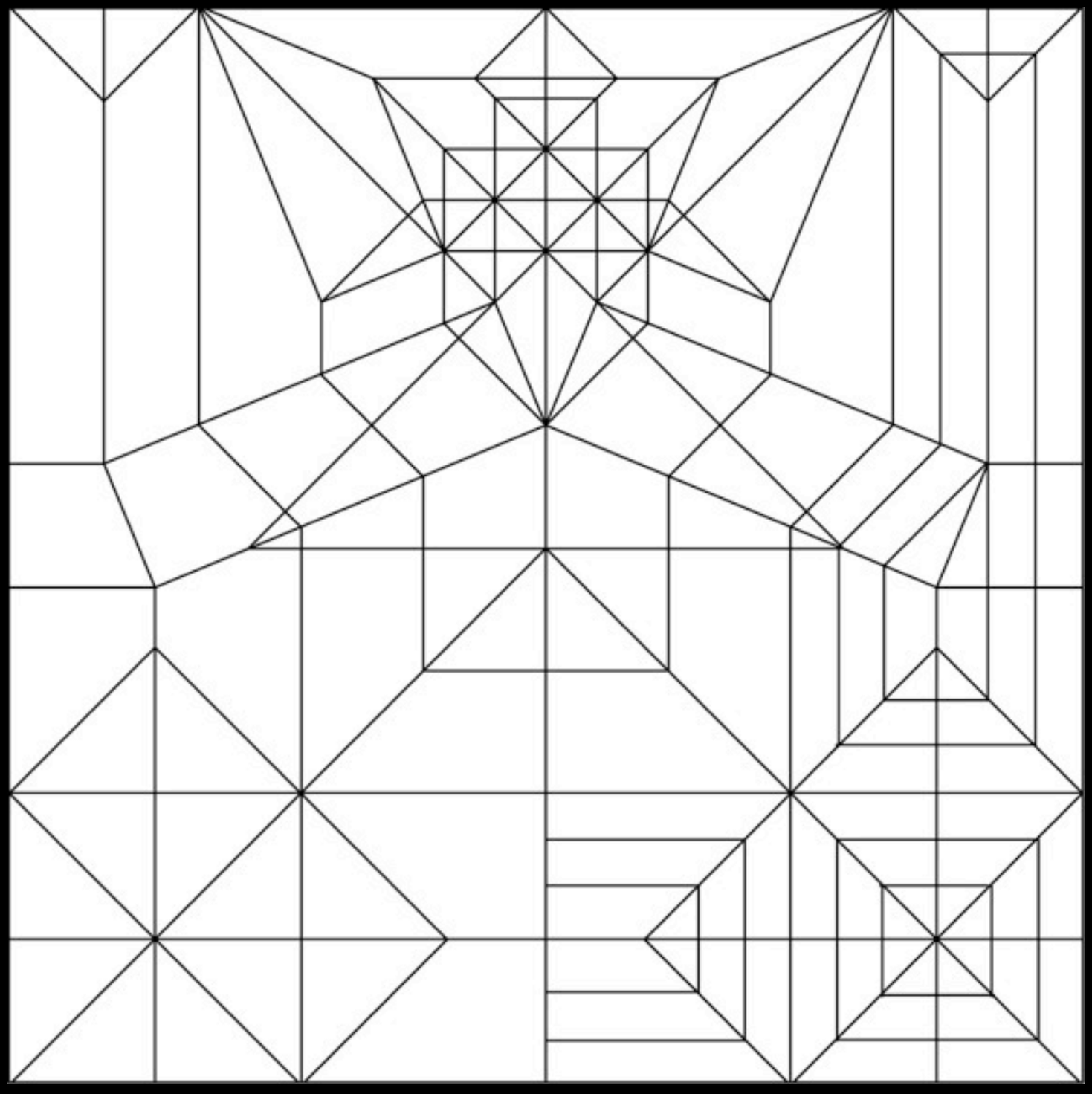
Problem: Optimizer can not find a solution due to trying to optimize under too many constraints
Solution: decrease the number of additional constraints

Example Files

- crab_book.tmd5 = crab with book symmetry
- crab_diag.tmd5 = crab with diagonal symmetry
- crab_book_tri.tmd5 = triangulated version of book
- crab_diag_tri.tmd5 = triangulated version of diagonal

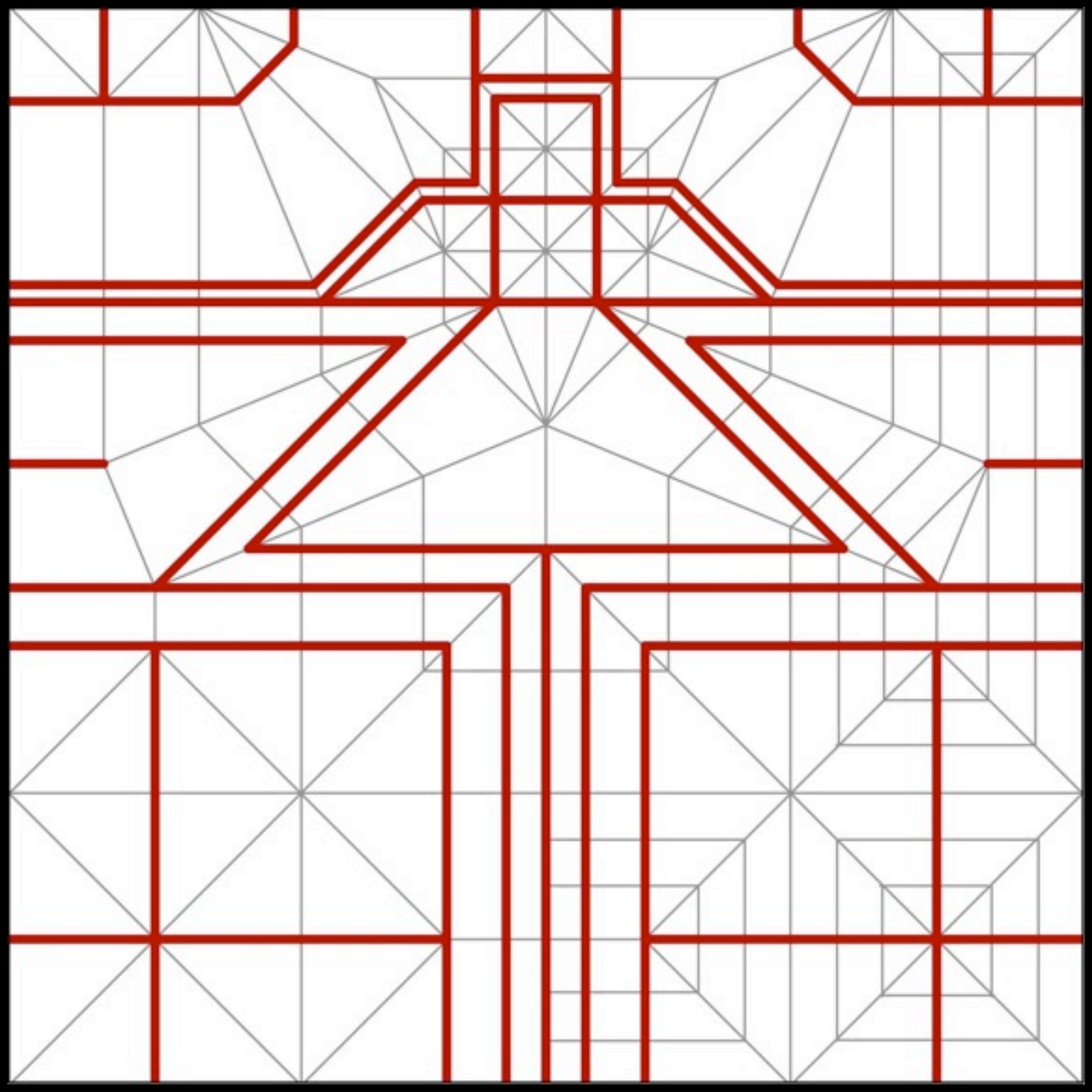


Non-TreeMaker Example



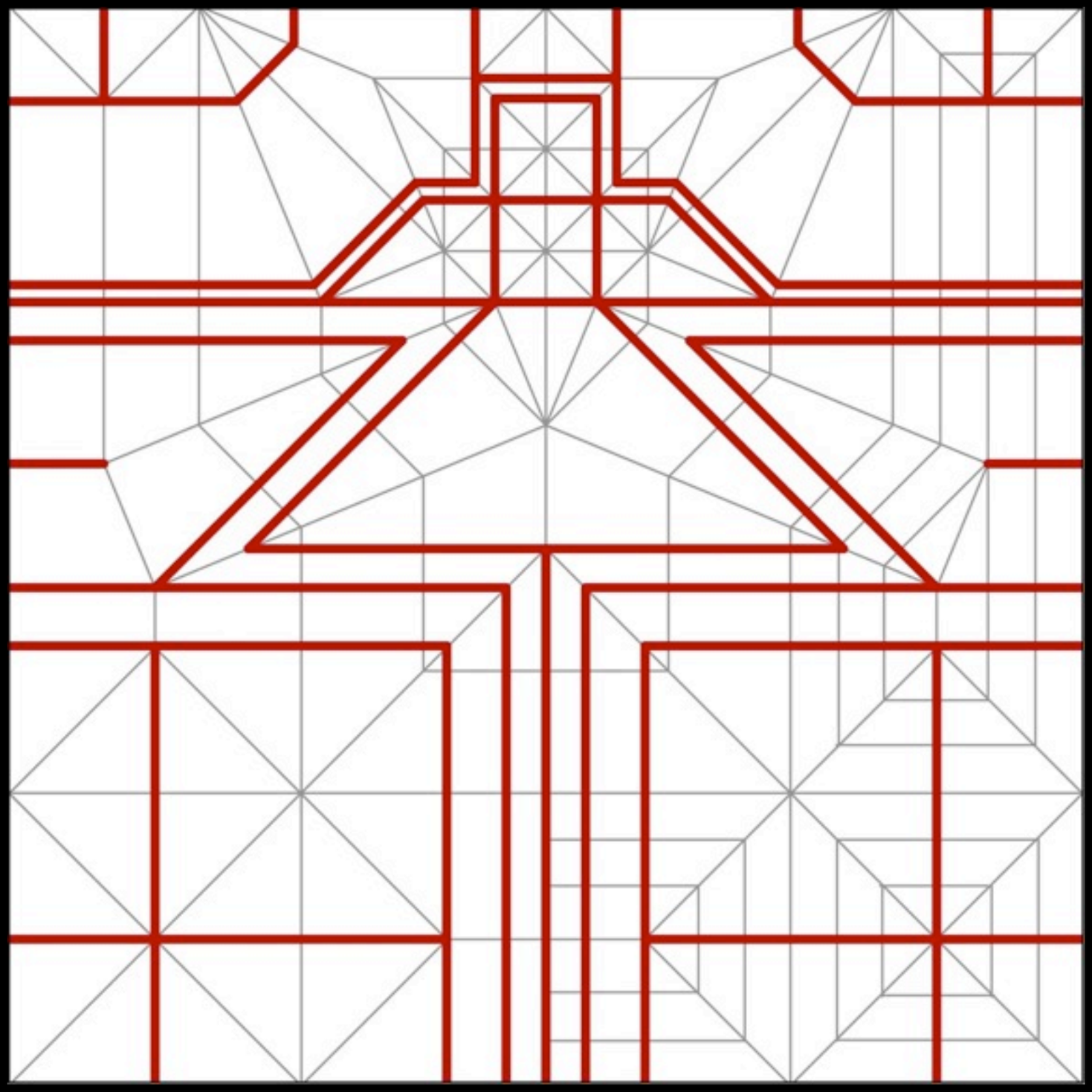
22.5 degree folding
Constrained under back geometry
Taking thickness into account
Non-uniaxial in ultimate folded form
Texture

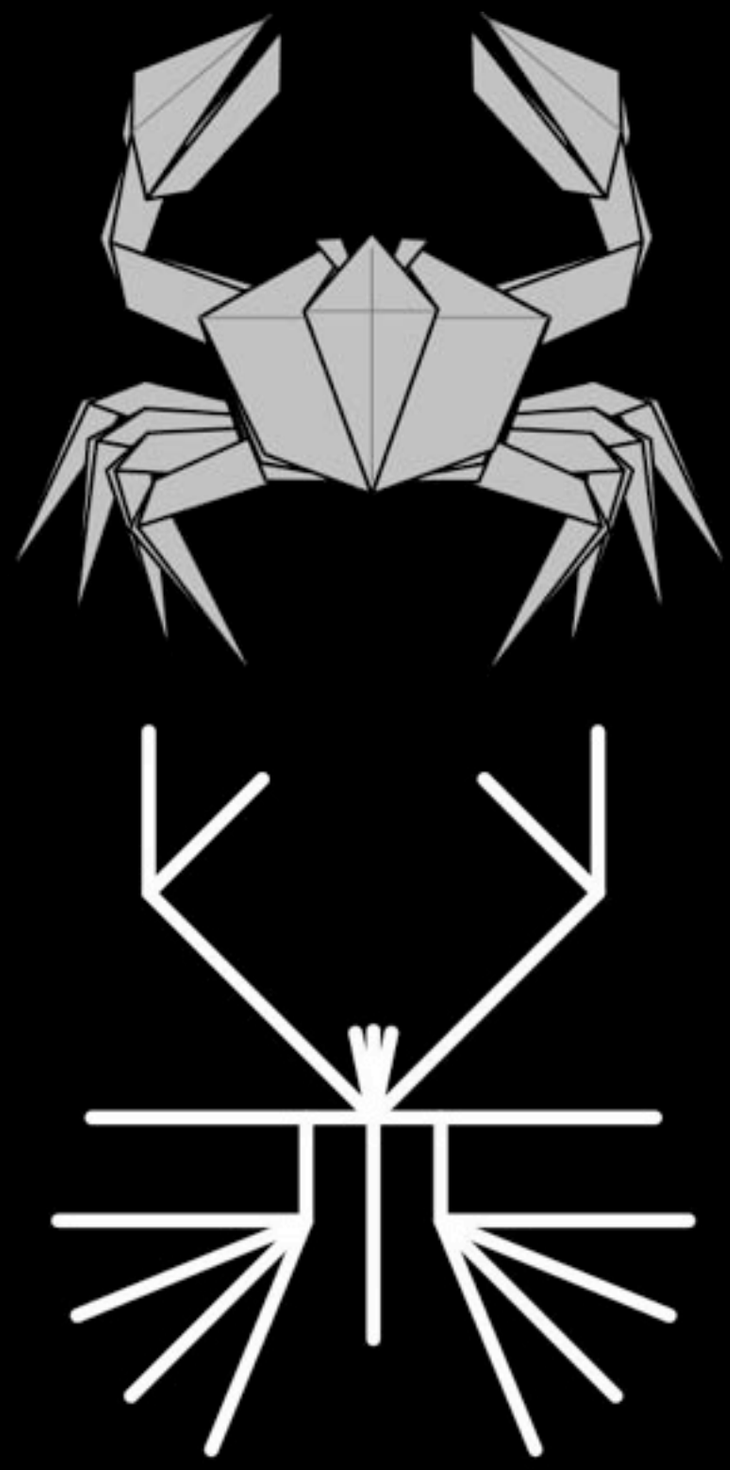
Non-TreeMaker Example



22.5 degree folding
Constrained under back geometry
Taking thickness into account
Non-uniaxial in ultimate folded form
Texture

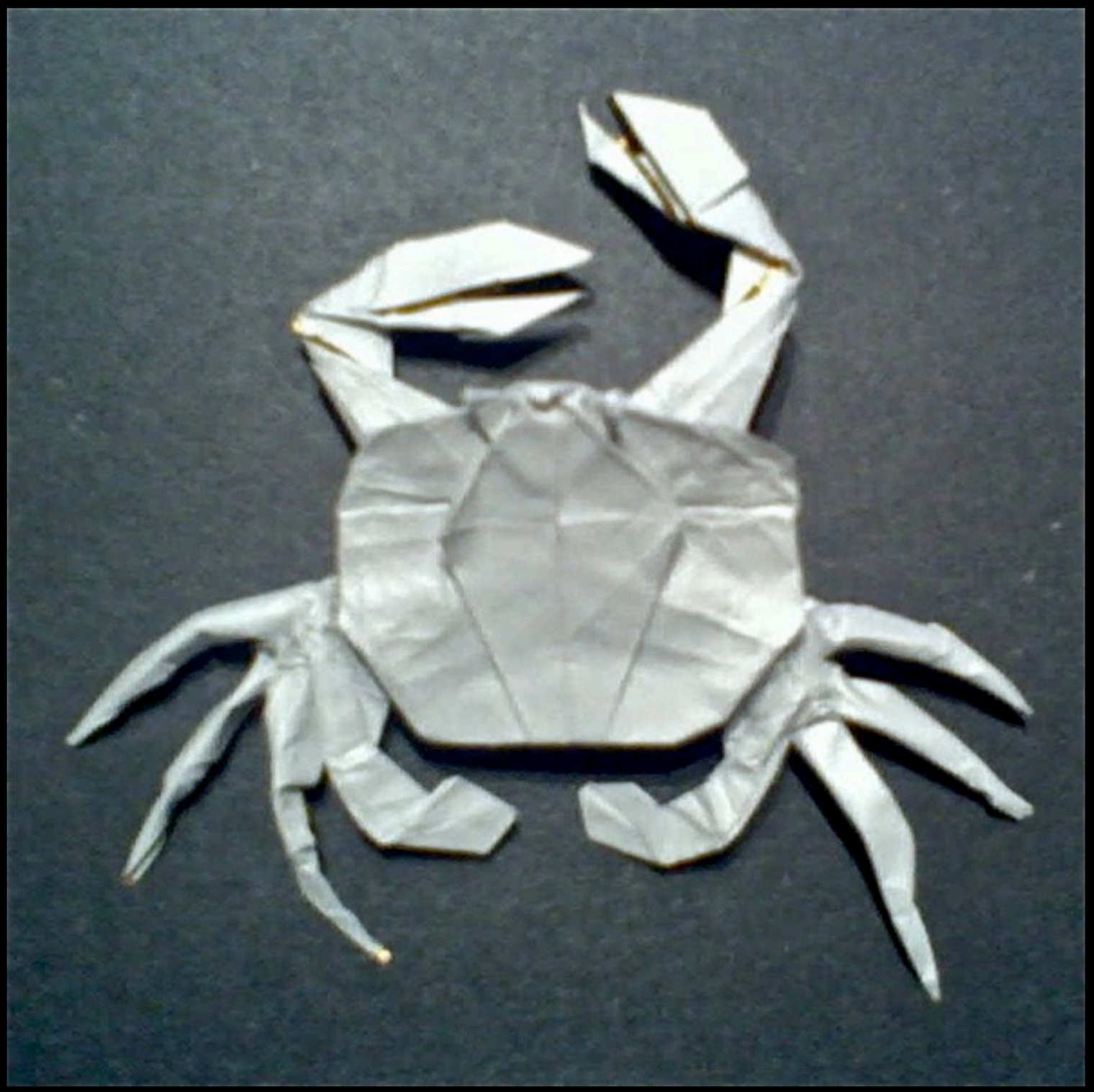
Non-TreeMaker Example





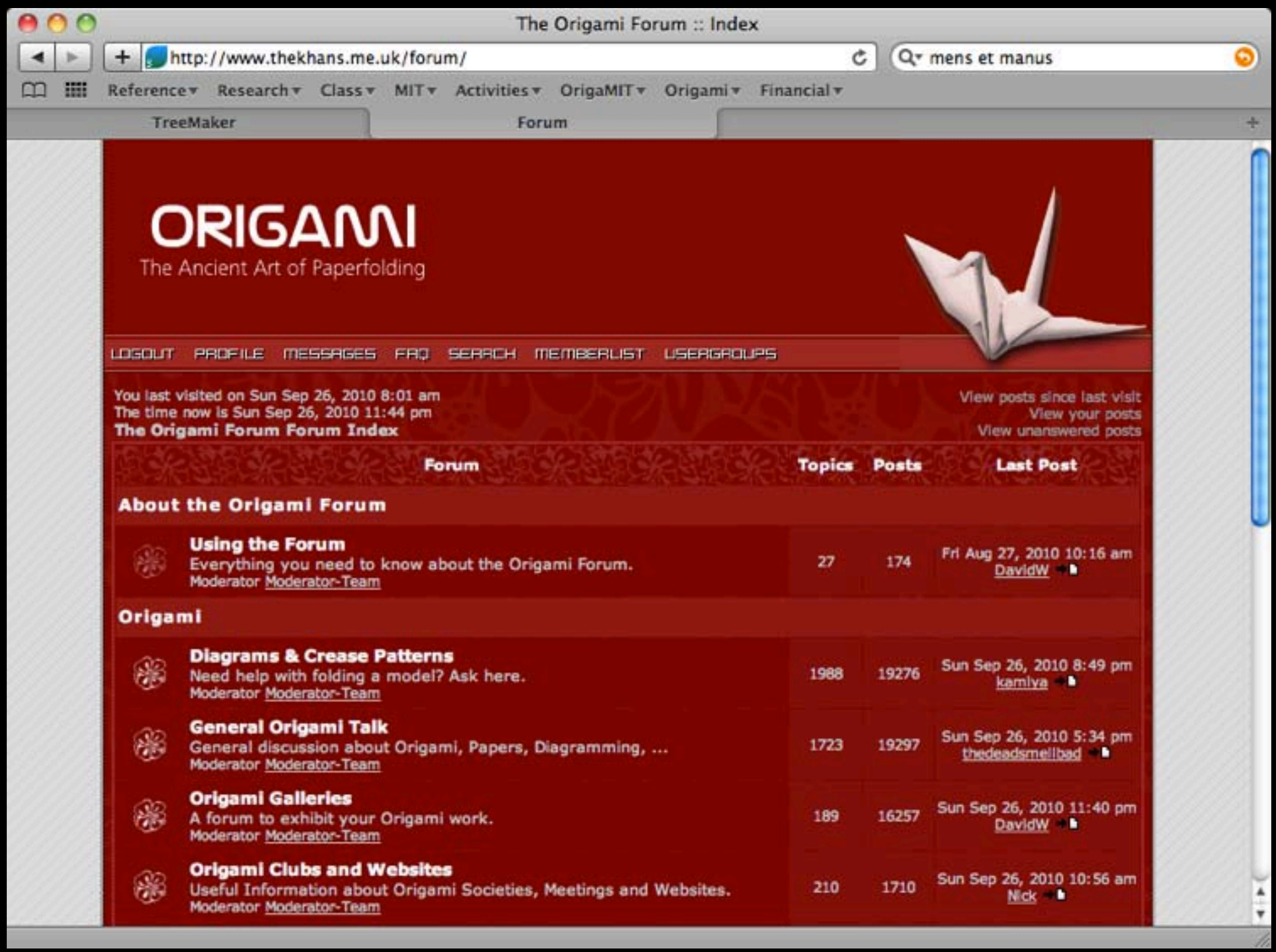
22.5 degree folding
Constrained under back geometry
Taking thickness into account
Non-uniaxial in ultimate folded form
Texture

Non-TreeMaker Example



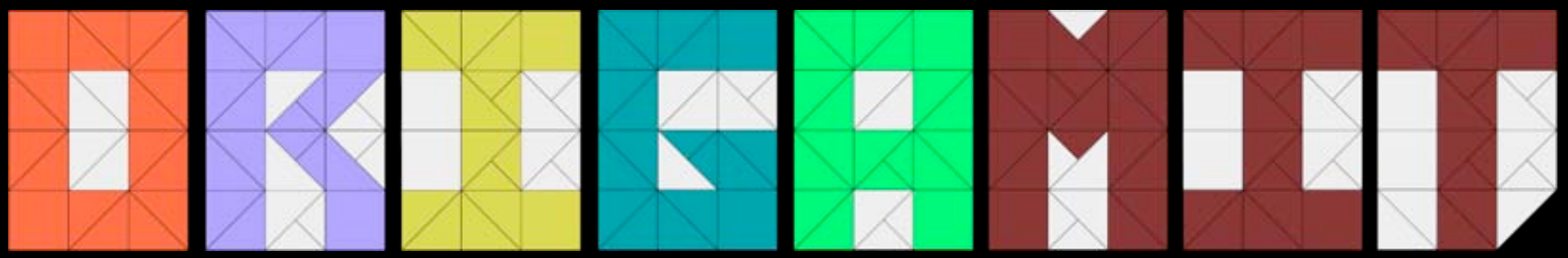
Proof of concept

Origami Forum



http://www.thekhans.me.uk/forum/

For more information on all things origami...



MIT's Origami Club

Weekly Meetings

Sundays 2-4pm
Student Center

http://origamit.scripts.mit.edu

Shameless promotion

6.849 Geometric Folding Algorithms: Linkages, Origami, Polyhedra

Fall 2012