Interactive Construction and Automated Proof in Eos System with Application to Knot Fold of Regular Polygons

Fadoua Ghourabi, Tetsuo Ida, and Kazuko Takahashi

1. Introduction

In origami geometry, the construction and the verification should go hand in hand. When we present a new origami by a new fold method, we will show certain geometric properties that enable us to claim its novelty by formal argument, i.e. proving. It is desirable to have some kind of automation by a computer towards computer-aided origami. Several systems have been implemented to simulate and treat complex origami constructions whereas proving in origami geometry remains in the hands of the constructor or someone well versed in geometrical theorem proving.

We have been developing a computational origami system with computational theorem proving capabilities, called Eos.[1] The engine of Eos consists of a solver, a graphical visualizer and a prover. The main functionality of the solver is to find a fold line by solving algebraic constraints. The properties that the fold line(s) should satisfy are described by a formula in a many-sorted first order language. The solver generates the algebraic interpretation of the formula that corresponds, in general, to a system of multi-variate polynomial equations, then solve them to determine suitable fold line(s) [**4**]. The graphical visualizer interacts with the solver and produces a graphical output for applying the fold along the line obtained by the solver. The visualizer uses a graph model of origami structure. The fold along a line is reduced to graph rewriting problem [**6**]. After the construction is completed, the origamist invokes the prover to prove the correctness of the construction. In other words, to prove geometrical properties of the origami object obtained at the end of the construction [**5**].

An overview of Eos system was given in Origami^4 [**7**]. Since then, Eos underwent several improvements. Its usability has been extended to solve and prove construction problems beyond Huzita's folds. In particular, the knot fold construction is an interesting example that exhibits some of the new features in Eos. The

This work was supported by JSPS KAKENHI Grant Number 25330007. The first author of this paper is supported by the postdoctoral fellowship at Kwansei Gakuin University.

[1] The most recent version of Eos requires an installation of Mathematica 9, and it can be downloaded from the webpage `http://www.i-eos.org/tutorial`.

use of knot fold to obtain regular polygons was studied in the past by few mathematicians (e.g. [**9, 2, 8**]). Making regular polygons by knot fold is a construction problem that can be fully tackled with Eos system, i.e. construction and proof of correctness. In this paper, we explain how knot fold is translated to a constraint solving problem. We show that Eos can express, solve and reason about the constraints. We illustrate with the examples of regular 2n+1-gons.

Eos is implemented on the top of *Mathematica* and follows its syntactical conventions. For the clarity of this paper, we will use the common notation for function call $f(x_1,\ \ldots,\ x_n)$ instead of *Mathematica*'s $f[x_1,\ \ldots,\ x_n]$. In parallel to the demonstration of construction examples, we will explain other elements of syntax when necessary.

The rest of the paper is organized as follows. In Sect. 2, we explain the representation of fold operations in Eos. In Sect. 3, we discuss the constraints that define a simple pentagonal knot. In Sect. 4, we present another alternative to define knot fold. We illustrate with the construction of regular heptagon. In Sect. 5, we show how we prove the correctness of knot fold construction using Eos. In Sect. 6, we summarize our results and point out directions of further research.

2. Fold Operation by Eos

The main operation in geometric origami is folding the paper along line(s). In Eos, fold operation is specified by a logical formula of the following form.

$$\exists_{x_1, x_1:\tau_1} \ldots \exists_{x_i, x_i:\tau_i}\ \phi_1(t_1, \ldots, t_k) \wedge \ldots \wedge \phi_s(t_1, \ldots, t_k) \tag{1}$$

The existentially quantified variables $x_1,\ \ldots,\ x_i$ are of sorts $\tau_1,\ \ldots\ \tau_i \in \{\mathrm{Line}, \mathrm{Point}, \mathrm{Num}\}$. The variables of sort Line denote the fold lines along which the folds are to be preformed. The variables of sort Point denote the points of intersections of fold lines and existing lines. The variables of sort Num denotes numbers.

$\phi_1(t_1, \ldots, t_k),\ \ldots,\ \phi_s(t_1, \ldots, t_k)$ are (positive or negative) atomic formulas over the geometric objects $t_1, \ldots, t_k$. Applying a fold operation is to find instances for $x_1,\ \ldots,\ x_i$ such that $\phi_1(t_1, \ldots, t_k) \wedge \ldots \wedge \phi_s(t_1, \ldots, t_k)$ holds, and then to fold the origami along the lines among $x_1,\ \ldots,\ x_i$.

Huzita's 6 fold operations (O1) - (O6) are written in the form of formula (1).[2]

$$\exists_{x, x:\mathrm{Line}}\ x = \mathrm{O}_i(t_1,\ \ldots, t_{k_i}),\quad \text{for } i = 1, 2, 4 \tag{2}$$

$$\exists_{x, x:\mathrm{Line}}\ \mathrm{O}_i(t_1,\ \ldots, t_{k_{i-1}}, x),\quad \text{for } i = 3, 5, 6 \tag{3}$$

Given the geometric objects $t_1, \ldots, t_{k_i}$, the function $\mathrm{O}_i(t_1, \ldots, t_{k_i})$ computes the fold line that satisfies operation (Oi), where $i = 1, 2, 4$. The equality predicate in $x = \mathrm{O}_i(t_1, \ldots, t_{k_i})$ is extended to lines. In (3), O_3, O_5 and O_6 are predicates, and not defined as a function that returns a fold line x since x may not be unique. Depending on $t_1, \ldots, t_{k_i}$, there are up to 2 possible fold lines in case of (O3) and (O5) and up to 3 fold lines in case of (O6). Hence, $\mathrm{O}_i(t_1, \ldots, t_{k_{i-1}}, x)$ is the predicate over geometric objects $t_1, \ldots, t_{k_i} (= x)$, and describes the superposition of points and lines in Huzita's fold operation (Oi), where $i = 3, 5, 6$. For instance, $\mathrm{O}_5(P, m, Q, x)$ states that a fold line x passing through point Q superposes point P and line m.

[2] We treat the six Huzita's basic fold operations he proposed in 1989, although one more presented later by Justin can be included for the exhaustive enumeration of basic fold operations that rely on incidence relations of point and lines.

Function HO (which stands for Huzita Ori) allows the orgamist to interact with Eos and perform a fold operation described by a formula of the form (1). Huzita's fold operations (O1) - (O6) are implemented in Eos and can be applied by evaluating HO with suitable arrangement of parameters. The call HO[P, m, Q] asks Eos to internally treat the formula $\exists_{x,x:\text{Line}}$ $\text{O}_5(P, m, Q, x)$, and solve for x that satisfies $\text{O}_5(P, m, Q, x)$.

The implementation of fold operation in Eos is extensible in the sense that the orgamist may ask the system to perform a fold operation beyond Huzita's fold operations. The organist can pass formula (1) to Eos as an argument of function HO.

$$\text{HO}((\exists_{x_1,x_1:\tau_1} \ldots \exists_{x_i,x_i:\tau_i} \phi_1(t_1, \ldots, t_k) \wedge \ldots \wedge \phi_s(t_1, \ldots, t_k)), \langle \textit{keyword arguments} \rangle)$$

Furthermore, the orgamist may need to do more than solving for $x_1, \ldots, x_i$. S/he can specify the names of the newly solved points, or tell Eos the direction of the fold along the line(s) among $x_1, \ldots, x_i$, i.e. mountain or valley, and so on. Such information is given as optional arguments in HO call of the form "keyword → value". Otherwise, Eos undertakes these tasks and apply default values.

3. Knot Fold of Regular Pentagon

The making of the simplest knot can be decomposed into the four steps shown in Fig. 1. We start with a rectangular origami or *origami tape* depicted in Fig. 1a. First, we make two folds as shown in Fig. 1b. Next, we take the end of the upper face and mountain-fold it while inserting it immediately above the bottom face. The result is shown in Fig. 1c. Finally, we pull the two ends of the folded tape to secure the knot and obtain a final shape of the regular pentagon in Fig 1d.

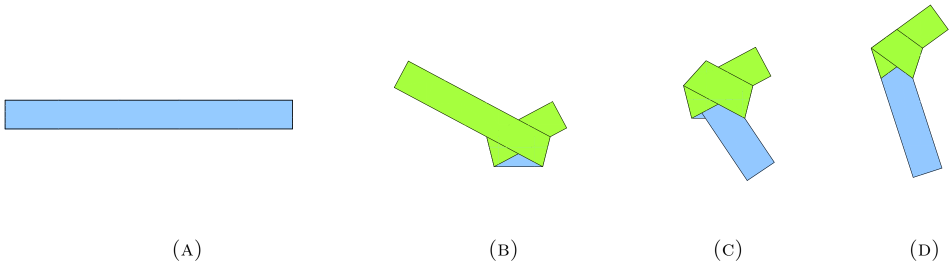

(A) (B) (C) (D)

Figure 1. The steps of making a simple origami knot

The three folds in Fig. 1b and 1c and the act of pulling the tape are obviously beyond Huzita's fold operations. The involved folds are mutually dependant, and can be regarded as a Alperin-Lang multi-fold operation [**1**]. We therefore specify the properties of the knot fold by a formula of the form (1). To specify them, we mark the key points of the knot, unfold it and examine the fold lines and the points that have been constructed as shown in Fig. 2.

The following shows the Eos program to construct the regular pentagon EHGKF in Fig. 2a by knot fold.

Program P1 [construction of a pentagonal knot]

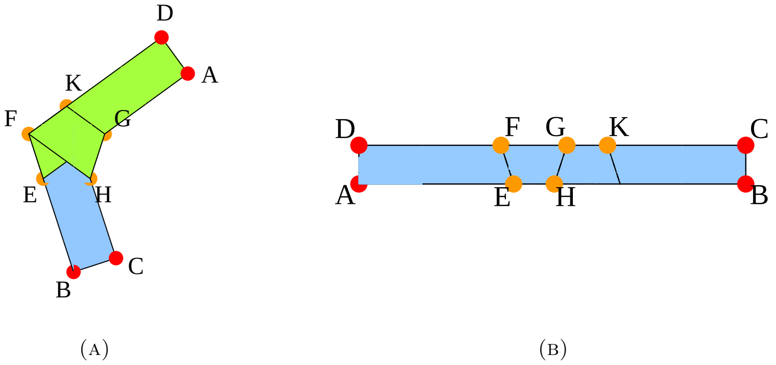


FIGURE 2. Unfolded knot fold of regular pentagon

1. BeginOrigami(Pentagonal knot, $\{100,\ 10\}$)
2. NewPoint($\{\text{E} \rightarrow \{40, 0\}\}$)
3. HO($\exists_{m,m:\text{Line}}\exists_{n,n:\text{Line}}\exists_{l,l:\text{Line}}\exists_{f,f:\text{Point}}\exists_{g,g:\text{Point}}\exists_{h,h:\text{Point}}\exists_{k,k:\text{Point}}$
   $(h \in \text{AB} \wedge \{f, g, k\} \subset \text{CD} \wedge f \in m \wedge h \in n \wedge k^n \in l \wedge$
   $\text{O5}(g, \text{EA}, \text{E}, m) \wedge \text{O5}(f, \text{EB}, g, n) \wedge \text{O5}(h, \overline{\text{C}^n g}, f, l) \wedge$
   $k^n \in \overline{\text{D}^m f} \wedge \text{E} \in \overline{f(\text{B}^n)^l}),$
   $\text{Case} \rightarrow 4, \text{MarkPointAt} \rightarrow \{\text{F}, \text{G}, \text{H}, \text{K}\}, \text{Handles} \rightarrow \{\text{A}, \text{B}, \text{B}\},$
   $\text{Direction} \rightarrow \{\text{Valley}, \text{Valley}, \text{Mountain}\}, \text{InsertFace} \rightarrow \{0, 0, \mathit{Bottom}\})$

Each line of the program is a call of EOS functions, i.e. the calls of BeginOrigami, NewPoint, and one HO. They can be evaluated separately but in the sequence of their appearance in Program P1.

Steps 1 and 2 are preparatory steps. At step 1, we start a new session of origami construction that we name "Pentagonal knot" with an initial origami of size 100×10. EOS defines a Cartesian coordinate system whose x-axis and y-axis are lines AB and AD, respectively. Initial points A, B, C and D are of coordinates (0, 0), (100, 0), (100, 10) and (0, 10), respectively. EOS uses this coordinate system to represent points as pairs of real numbers (Cartesian coordinates) and lines and curves as polynomial equations. In particular, a line m is represented by the equation $ax + by + c = 0$.

At step 2, let E be an arbitrary points on the line AB. For the sake of the construction, we mark a point E at (40, 0).

At step 3, we apply the geometric construction described in the formula first argument of HO. By solving the constraint, EOS returns three fold lines, i.e. m, n and l and four points f, g, h and k. Note that points F, G, H and K in Fig. 2a are solutions for variables f, g, h and k. Referring to Fig. 2b, we establish the incidence relations between points and lines involved in the knot fold, i.e. $h \in \text{AB} \wedge \{f, g, k\} \subset \text{CD} \wedge f \in m \wedge h \in n \wedge k^n \in l$. We abuse the set notation, and use $X \in m$ to mean that point X is incident to line m, and $\{X_1, \ldots, X_k\} \subset m$ to mean that all the points $X_1, \ldots, X_k$ are incident to line m. Given a point X and a line t, X^t denotes the reflection of point X across line t. Note that variable k corresponds to the location of point K before knotting, i.e. point K in Fig. 2b, whereas point k^n in $k^n \in l$ corresponds to the location of point K after knotting, i.e. Fig. 2a.

As indicated in the sub formula $\mathrm{O5}(g, \mathrm{EA}, \mathrm{E}, m) \wedge \mathrm{O5}(f, \mathrm{EB}, g, n) \wedge \mathrm{O5}(h, \overline{\mathrm{C}^n g}, f, l)$, we perform three (O5) operations. Regarding the third (O5), we first explain how to read Eos's notation $\overline{\mathrm{C}^n g}$ in $\mathrm{O5}(h, \overline{\mathrm{C}^n g}, f, l)$. We write $\overline{XY}$ to mean the line passing through points X and Y. Hence, $\overline{\mathrm{C}^n g}$ is the line passing through points C^n and g. The fold along line l passing through f superposes point h and line $\overline{\mathrm{C}^n g}$.

For any point f on CD, we can perform the three (O5) operations, and, hence, there are infinite solutions for the above properties. We see that the shape in Fig. 3 results from the three (O5) operations. In practice, we need to pull the paper until all points reach their "proper" locations in Fig. 2a. The immediately noticed difference in the shape of Fig. 2a w.r.t. the one in Fig. 3 is that points K and E are incident to lines FD and FB, respectively. We therefore add the following incidence constraint $k^n \in \overline{\mathrm{D}^m f} \wedge \mathrm{E} \in \overline{f(\mathrm{B}^n)^l}$.

By solving the constraint, Eos returns three fold lines and four points. However, there are four distinct solutions. The argument "Case → 4" is added to HO to choose the solution that leads to a regular pentagon, i.e. the 4th solution. The solutions for variables f, g, h and k are given the names F, G, H and K which is specified by argument "MarkPointAt → $\{\mathrm{F}, \mathrm{G}, \mathrm{H}, \mathrm{K}\}$". By this way, the points bound to the existential variables become available in the following steps of the construction. The keyword argument "Handles → $\{\mathrm{A}, \mathrm{B}, \mathrm{B}\}$" determines which side of the fold lines to be moved. In this case, the face that is to the left to the fold line m, i.e. the face containing point A, is moved by fold. The face that is to the right of fold line n is moved. The face that is to the left of the fold line l is moved. "Direction → $\{\mathrm{Valley}, \mathrm{Valley}, \mathrm{Mountain}\}$" asks Eos to perform valley folds along lines m and n and a mountain fold along line l. "InsertFace → $\{0, 0, \mathrm{Bottom}\}$" is to insert the moving faces above (below in the case of the valley fold) the face in the list. The argument may be a list of faces for the same reason of Direction. Symbol 0 in the list means "no effect". The outputs of the above HO call are shown in Fig. 4.

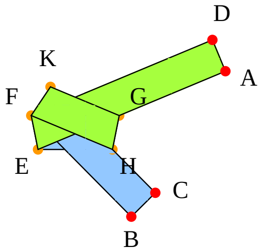
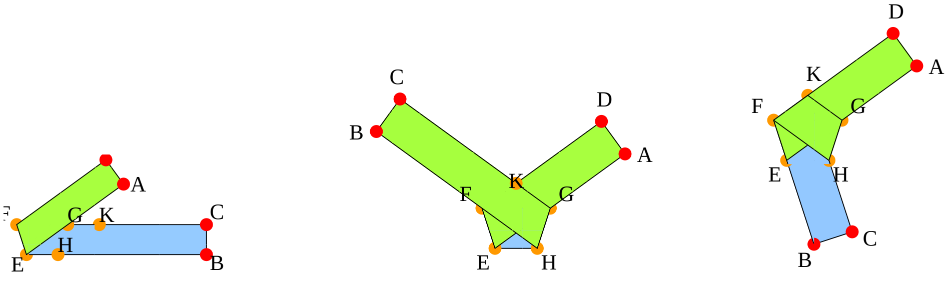


FIGURE 3. The knot fold before pulling the tape

4. Knot Fold of Regular Heptagon

We now examine the knot fold from an algebraic point of view. Through the example of regular heptagon, we show another alternative to define the constraints on knot fold using polynomial equations and point equalities. Starting from an initial tape ABCD and a point E on the line AB, we will construct the regular heptagon ELHGJKF in Fig. 5. The algebraic constraints specified on points F, H, G, J, K and L are written as a formula of form (1).



(A) (B) (C)

Figure 4. Construction of regular pentagon EFKGH

$$\exists_{m,m:\mathrm{Line}}\exists_{n,n:\mathrm{Line}}$$

$$\exists_{f,f:\mathrm{Point}}\exists_{h,h:\mathrm{Point}}\exists_{g,g:\mathrm{Point}}\exists_{j,j:\mathrm{Point}}\exists_{k,k:\mathrm{Point}}\exists_{l,l:\mathrm{Point}}$$

$$\exists_{ht,ht:\mathrm{Num}}\exists_{p,p:\mathrm{Num}}\exists_{q,q:\mathrm{Num}}\exists_{r,r:\mathrm{Num}}$$

$$(\{f,g,j,k\}\subset \mathrm{CD}\wedge\{h,l\}\subset\mathrm{AB}\wedge\{\mathrm{E},f\}\subset m\wedge\{h,g\}\subset n\wedge \tag{4}$$

$$f-\mathrm{E}=\mathrm{Point}[-ht\times p,ht]\wedge g-h=\mathrm{Point}[ht\times p,ht]\wedge \tag{5}$$

$$h-\mathrm{E}=\mathrm{Point}[2ht\times r\times q,0]\wedge k-g=\mathrm{Point}[2ht\times r\times q,0]\wedge \tag{6}$$

$$j-f=\mathrm{Point}[2ht\times r\times q]\wedge \mathrm{E}-l=g-j\wedge \tag{7}$$

$$p^2+1=q^2\wedge p=(4r^3-3r)q\wedge 8r^3-4r^2-4r+1=0) \tag{8}$$

The existentially quantified lines m and n, points f, g, h, j, k and l, and variables ht, p and q satisfy the constraints (4) - (8). Equation (4) shows the relation of incidences between points f, g, j, k and l and lines m, n, AB and CD. Equations (5) - (7) express the location of points f, g, h, j, k and l with respect to the location of point E in the following way.

- The interior angles of ELHGJKF are equal to $\frac{5\pi}{7}$, in particular $\measuredangle\mathrm{LEF}=\frac{5\pi}{7}$. Let $\theta=\frac{\pi}{7}$. We deduce that $\measuredangle\mathrm{AEF}=3\theta$ and $\measuredangle\mathrm{HEL}=\theta$. The slope of the fold line EF is, therefore, equal to $-\tan(3\theta)$. Furthermore, let p, q and r be three variables of type real satisfying $\frac{p}{q}=\cos(3\theta)$ and $r=\cos(\theta)$. We construct the perpendicular FX to line AB passing through F and whose foot is point X on AB. Let ht be the hight of the tape, i.e. $ht=|\mathrm{AD}|$. We infer that $|\mathrm{FX}|$, $|\mathrm{XE}|$ and $|\mathrm{EF}|$ are equal to ht, $ht\times p$ and $ht\times q$, respectively. Similarly, we can infer that $|\mathrm{GY}|$, $|\mathrm{HY}|$ and $|\mathrm{HG}|$ are equal to ht, $ht\times p$ and $ht\times q$, where line HY is the perpendicular to line CD and whose foot is point Y on CD. The operators minus and plus are extended to points. The expression $X-Y$ is a coordinate-wise subtraction. Thus, in (5), we have $f-\mathrm{E}=\mathrm{Point}[-ht\times p,ht]$ and $g-h=\mathrm{Point}[ht\times p,ht]$.
- In order to determine the location of point H, we consider the isosceles triangle $\triangle$LHE. It is straightforward to see that the slope of line EL is equal to $-\tan(\theta)$. Let LZ be the perpendicular to line AB whose foot is point Z on AB. We have

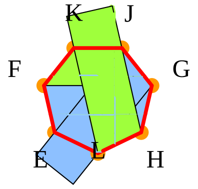


FIGURE 5. Knot fold of regular heptagon

that |EZ| = |EF| × r, and we deduce that |EH| = $2ht \times q \times r$. The same property holds for the isosceles triangles △JGK and △KFJ, and hence we deduce the equalities in (6) and (7).

Regarding the polynomial equalities in (8), recall that $p = q \times \cos(3\theta)$. From trigonometric laws, we have $p = q(4r^3 - 3r)$ and $p^2 + 1 = q^2$. The number r (i.e. $\cos(\theta)$) is a solution of the cubic equation $8r^3 - 4r^2 - 4r + 1 = 0$. Hence, p, q and r satisfy the equations $\{p^2 + 1 = q^2,\ p = (4r^3 - 3r)q,\ 8r^3 - 4r^2 - 4r + 1 = 0\}$.

Function HO solves the algebraic constraints and yields to 6 distinct sets of solutions. Each solution set includes the coefficients of lines m and n, the coordinates of points F, G, H, J, K and L (when the knot is unfolded) and the values of numbers p, q and r. Only for explanation purposes, we manually compute and draw the final coordinates of points F, G, H, J, K and L as well as the edges of the desired regular heptagon. We obtain the 6 cases depicted in Fig. 6. Equation $8r^3 - 4r^2 - 4r + 1 = 0$ has three distinct solutions of the form $\cos(n\theta)$, where $n = 1, 3, 5$. The regular heptagon in Figs. 6a and 6f corresponds to the solution $\cos(\theta)$, the star polygons in Figs. 6b and 6c to the solution $\cos(3\theta)$, and the star polygons in Figs. 6d and 6e to the solution $\cos(5\theta)$. Since point H is on line AB, it can be either of the half-line EA or the half-line EB, which explains the symmetry of the solutions. The choice of the 6th case that corresponds to Fig. 6f leads to the tegular heptagon ELHGJKF in Fig. 5.

5. Correctness of Knot Fold

After an origami construction is completed, we prove its correctness. While the origami construction in EOS is interactive, the proof of the correctness is automated. EOS system is in the category of systems that employ automated proving methods based on algebraic algorithms, i.e. Gröbner basis and cylindrical algebraic decomposition.

5.1. Proof of Correctness in Eos. Proving in EOS is to decide that a relevant geometric property, called *conclusion* or *goal*, follows from a collection of geometric hypothesis, called *premise*. In other word, to show that the formula premise ⇒ conclusion is satisfiable by a program. In EOS, the premise is the conjunction of logical formulas specified in HO calls that we denote by $\mathcal{P}$. The formula $\mathcal{P}$ is internally recorded during the construction. The conclusion is the claim that certain geometric property holds for the constructed shape, e.g. the regularity of the constructed shape in the case of polygonal knot fold. The conclusion is specified by the orgamist as a logical formula and is passed to EOS for internal treatment. We use $\mathcal{C}$ to denote the conclusion formula.

Depending on the algebraic interpretation of $\mathcal{P}$ and $\mathcal{C}$, Eos decides which algorithm to employ. In the case that only equalities (and disequalities) are involved in the algebraic forms, Eos uses Gröbner basis computation. If inequalities are involved, the algebraic algorithm used by Eos is cylindrical algebraic decomposition. In both cases, Eos uses *Mathematica*'s built-in functions for computing Gröbner basis and cylindrical algebraic decomposition. When the computation terminates, Eos generates a ProofDoc that describes the details of the construction and the proof [**3**].

Since only polynomial equalities are involved in the knot fold constructions described in Sect. 3 and Sect. 4, the proof in Eos employs Gröbner basis computation. The proof is by contradiction. Namely, in order to show that $\mathcal{P} \Rightarrow \mathcal{C}$ holds, we prove that $\neg(\mathcal{P} \Rightarrow \mathcal{C})$, which is logically equivalent to $\mathcal{P} \wedge \neg\mathcal{C}$, doesn't hold. This is algebraically formalized as ideal membership problem. Eos computes one significant ideal generated by the polynomials equalities in the algebraic interpretation of $\mathcal{P} \wedge \neg\mathcal{C}$, namely the reduced Gröbner basis that we denote by $\mathrm{GB}(\mathcal{P} \wedge \neg\mathcal{C})$. If $1 \in \mathrm{GB}(\mathcal{P} \wedge \neg\mathcal{C})$, then the polynomial equalities that describe $\mathcal{P} \wedge \neg\mathcal{C}$ have no common solution which means that the proposition $\mathcal{P} \wedge \neg\mathcal{C}$ is false. Hence, the proposition $\mathcal{P} \Rightarrow \mathcal{C}$ is true.

In the next two sections, we explain the proof of the correctness of knot fold of regular pentagon and regular heptagon. To that end, we show that the constructed shape in Fig. 4c and Fig. 5 are regular.

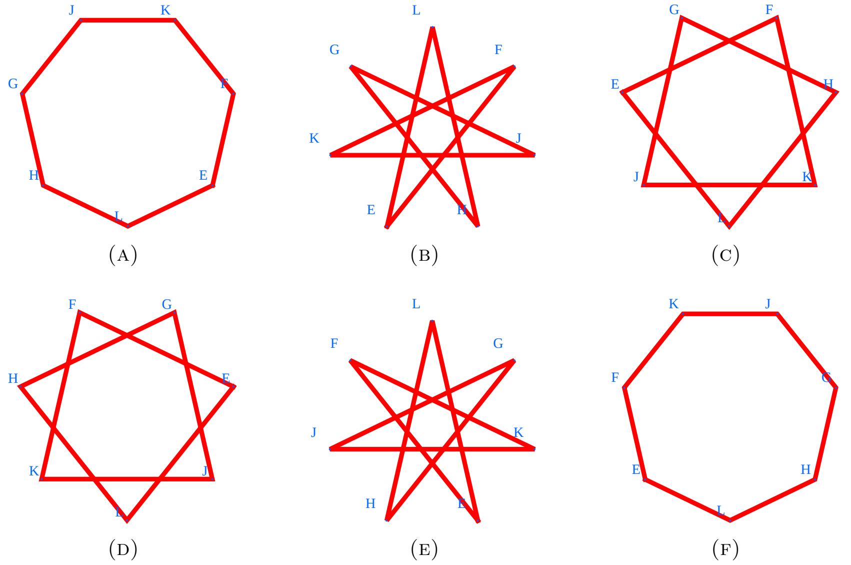


FIGURE 6. All the cases generated by function HO

5.2. Proof of Knot Fold of Regular Pentagon.

We prove the correctness of the knot fold of regular pentagon by showing that EFKGH in Fig. 4c is regular. To that end, we prove that the edges of EFKGH are symmetric by a rotation of an angle θ equal to $\frac{\pi}{5}$. We use Eos's function Goal to specify this property of symmetry.

$$\begin{aligned}&\mathrm{Goal}(\forall_{\alpha,\alpha:\mathbb{C}}\ (\alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{EF}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{FK}}) = 0 \Rightarrow\\ &\qquad \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{FK}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{KG}}) = 0\ \wedge \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{KG}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{GH}}) = 0 \wedge\\ &\qquad \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{GH}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{HE}}) = 0\ \wedge\ \alpha^5 - 1 = 0))\end{aligned}$$

Function ToZ($\overrightarrow{XY}$) computes the complex number $(v-u)+\imath\ (w-s)$, where points X and Y are of coordinates $(u,\ s)$ and $(v,\ w)$, respectively. Hence, α ToZ($\overrightarrow{XY}$) is the rotation of vector $\overrightarrow{XY}$ w.r.t an angle θ such that $\alpha = \cos(\theta) + \imath \sin(\theta)$.

Recall that Eos internally generates the algebraic interpretation of $\mathcal{P} \wedge \neg\mathcal{C}$, where $\mathcal{P}$ is conjunction of formulas accumulated during the knot fold construction and $\mathcal{C}$ is the formula argument of the above call of function Goal. We call function Prove to ask Eos to compute GB($\mathcal{P} \wedge \neg\mathcal{C}$).

Prove("Regular knot pentagon",
Mapping → {A → $\{0,0\}$, B → $\{1,0\}$, C → $\{1,1\}$, D → $\{0,1\}$, E → $\{w,0\}$}
Tactics → {Subgoal → SquaredDistance(E, F)=SquaredDistance(H, G)})

The first parameter of the function call of Prove is the label naming the proposition to be proved, and the second parameter is a list of the initial point mapping. Without loss of generality, we let the height of the initial origami to be 1. Point E is taken to be arbitrary on the edge AB. The mapping attributes the coordinates (w, 0) to point E. Variable w is arbitrary and treated by Eos as independent variable. This mapping is used to prove $\mathcal{P} \Rightarrow \mathcal{C}$ in the general case, i.e. for any point E on line AB.

The keyword argument "Tactics" introduces a set of proof tactics. In the above call of Prove, we ask Eos to use an extra subgoal. Eos first proves a useful intermediate lemma about equality of distances. In particular, a fundamental result in knot fold is showing that when folding a tape along a line non-parallel to the edges, we make an isosceles triangle [**9**]. The first two folds in Fig. 1b are about making two congruent isosceles triangles △GFE and △FHG. Therefore, we have |EF| = |HG|. Note that the equality of distances is squared.

The subgoal "SquaredDistance(E, F) = SquaredDistance(H, G)", denoted by $\mathcal{E}$, can be deduced from $\mathcal{P}$. Namely, Eos shows that the two formulas $\mathcal{P} \Rightarrow \mathcal{E}$ and $\mathcal{P} \wedge \mathcal{E} \Rightarrow \mathcal{C}$ hold by checking that $1 \in$GB($\mathcal{P} \wedge \neg\mathcal{E}$) and $1 \in$GB($\mathcal{P} \wedge \mathcal{E} \wedge \neg\mathcal{C}$). The introduction of the subgoal is not necessary but has the advantage of considerably speeding up the computation of Gröbner basis.

5.3. Proof of Knot Fold of Regular Heptagon.

Similarly to the proof of knot fold of regular pentagon, we specify a logical formula for the conclusion, and pass it to Eos.

$$\begin{aligned}&\mathrm{Goal}(\forall_{\alpha,\alpha:\mathbb{C}}\ (\alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{EF}}) - Toz(\overrightarrow{\mathrm{FK}}) = 0 \Rightarrow\\ &\qquad \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{FK}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{KJ}}) = 0\ \wedge \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{KJ}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{JG}}) = 0 \wedge\\ &\qquad \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{JG}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{GH}}) = 0\ \wedge \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{GH}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{HL}}) = 0 \wedge\\ &\qquad \alpha\ \mathrm{ToZ}(\overrightarrow{\mathrm{HL}}) - \mathrm{ToZ}(\overrightarrow{\mathrm{LE}}) = 0\ \wedge \alpha^7 - 1 = 0)\end{aligned}$$

We call function Prove to prove that EFKJGHGL is a regular heptagon for arbitrary point E on edge AB.

Prove("Regular knot heptagon",
    Mapping → $\{\mathrm{A} \to \{0,0\}, \mathrm{B} \to \{1,0\}, \mathrm{C} \to \{1,1\}, \mathrm{D} \to \{0,1\},$
    $\mathrm{E} \to \{w,0\}\}$, Tactics→{Split})

Note that conclusion $\mathcal{C}$ is of the form $\mathcal{C}_1 \Rightarrow \mathcal{C}_2 \wedge \ldots \wedge \mathcal{C}_7$ whose negation is equivalent to $(\mathcal{C}_1 \wedge \neg\mathcal{C}_2) \vee \ldots \vee (\mathcal{C}_1 \wedge \neg\mathcal{C}_7)$. By writing "Tactics→{Split}", we ask Eos to split the proposition $\mathcal{P} \wedge ((\mathcal{C}_1 \wedge \neg\mathcal{C}_2) \vee \ldots \vee (\mathcal{C}_1 \wedge \neg\mathcal{C}_7))$ into separate formulas $\mathcal{P} \wedge (\mathcal{C}_1 \wedge \neg\mathcal{C}_i)$, where $2 \leq i \leq 7$. Eos independently computes each of the $\mathrm{GB}(\mathcal{P} \wedge (\mathcal{C}_1 \wedge \neg\mathcal{C}_i))$, where $2 \leq i \leq 7$. Solving the knot fold constraints give rise to the 6 solutions depicted in Fig. 6. Although in the construction we choose a specific solution, i.e. Fig. 6f, the property of regularity is proved for all the solutions. The proof completes with the message "Proof by Groebner basis method is successful", i.e. $1 \in \mathrm{GB}(\mathcal{P} \wedge (\mathcal{C}_1 \wedge \neg\mathcal{C}_i)$, for all $2 \leq i \leq 7$.

6. Conclusion

We presented the construction of knot folds of regular pentagon and regular heptagon using constraint solving. We further showed the proof of the correctness of the construction of a regular heptagon based on Gröbner bases. From our experience with proving in origami geometry, working with Gröbner bases represent several challenges that we could tackle as further research.

References

1. R. C. Alperin and R. J. Lang, *One-, Two, and Multi-fold Origami Axioms*, Origami^4, Proceedings of the Fourth International Meeting of Origami Science, Mathematics, and Education (4OSME), 2009, pp. 371–393.
2. J. K. Brunton, *Polygonal Knots*, The Mathematical Gazette **45** (1961), no. 354, 299–301.
3. F. Ghourabi, T. Ida, and A. Kasem, *Proof Documents for Automated Origami Theorem Proving*, Automated Deduction in Geometry, LNCS, vol. 6877, Springer, 2011, pp. 78–97.
4. F. Ghourabi, T. Ida, H. Takahashi, M. Marin, and A. Kasem, *Logical and Algebraic View of Huzita's Origami Axioms with Applications to Computational Origami*, Proceedings of the 22nd ACM Symposium on Applied Computing (SAC'07) (Seoul, Korea), 2007, pp. 767–772.
5. T. Ida, A. Kasem, F. Ghourabi, and H. Takahashi, *Morley's Theorem Revisited: Origami Construction and Automated Proof*, Journal of Symbolic Computation **46** (2011), no. 5, 571 – 583.
6. T. Ida and H. Takahashi, *Origami Fold as Algebraic Graph Rewriting*, Journal of Symbolic Computation **45** (2010), 393–413.
7. T. Ida, H. Takahashi, M. Marin, A. Kasem, and F. Ghourabi, *Computational Origami System Eos*, Proceedings of 4th International Conference on Origami, Science, Mathematics and Education, 4OSME, 2006, Caltech, Pasadena CA, p. 69.
8. J. Maekawa, *Introduction of the Study of Knot Tape*, Origami^5, Proceedings of the Fourth International Meeting of Origami Science, Mathematics, and Education (5OSME), CRC Press, 2011, pp. 395–403.
9. K. Sakaguchi, *On Polygons Made by Knotting Slips of Paper*, Technical Report of Research Institute of Education, Nara University of Education **18** (1982), 55–58.

Fadoua Ghourabi, Department of Informatics, Kwansei Gakuin University, Japan
E-mail address: ghourabi@kwansei.ac.jp

Tetsuo Ida, Professor Emeritus, University of Tsukuba, Japan
E-mail address: ida@cs.tsukuba.ac.jp

Kazuko Takahashi, Department of Informatics, Kwansei Gakuin University, Japan
E-mail address: ktaka@kwansei.ac.jp