Unspecified Book Proceedings Series

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-assisted 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 geometric theorem proving.

We have been developing a computational origami system with computational theorem proving capabilities, called Eos. A description of earlier version of Eos system was given in 4OSME [**8**]. 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 use of knot fold to make regular polygons was studied by few mathematicians (e.g. [**2, 10, 9**]). 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 for Eos. We show that Eos can express, solve and reason about the constraints. We show this with the examples of regular $2n$+1-gons.

The rest of the paper is organized as follows. In Sect. 2, we present Eos system. In Sect. 3, we discuss the constraints of pentagonal knot. In Sect. 4, we present another method of defining 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.

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.

2. Eos System

2.1. Brief Overview. 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 [**7**]. After the completion of the construction, the origamist invokes the prover to prove the correctness of the construction. In other words, to prove geometric properties of the origami object obtained at the end of the construction [**6**].

Eos is implemented on the top of *Mathematica* and follows its syntactical conventions.[1] Due to space limitation, we only explain the elements of syntax that are used in this paper. 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 Eos, the set notation is extended to the incidence relation between points and lines. Expression $X \in m$ means that point X is incident to line m, and $\{X_1,\ \ldots,\ X_k\} \subset m$ means that all the points $X_1,\ \ldots,\ X_k$ are incident to line m. The reflection of point X across a line t is denoted by X^t in Eos. To refer to a line passing through points X and Y, we use $\overline{XY}$, or XY if the denotation is clear from the context.

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

$$\exists_{x_1,x_1:\tau_1} \ldots \exists_{x_i,x_i:\tau_i}\ \phi_1(t_{1,1},\ \ldots,\ t_{1,k_1}) \wedge \ldots \wedge \phi_s(t_{s,1},\ \ldots,\ t_{s,k_s}) \tag{1}$$

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.

$\phi_1(t_{1,1}, \ldots, t_{1,k_1}),\ \ldots,\ \phi_s(t_{s,1}, \ldots, t_{s,k_s})$ are literals over the geometric objects $t_{1,1},\ \ldots, t_{s,k_s}$. When we apply a fold operation, we first find instances for $x_1,\ \ldots,\ x_i$ such that $\phi_1(t_{1,1}, \ldots, t_{1,k_1}) \wedge \ldots \wedge \phi_s(t_{s,1}, \ldots, t_{s,k_s})$ holds, and then fold the origami along the lines $x_1,\ \ldots,\ x_i$. Huzita's six fold operations (O1) $\sim$ (O6) [**5**] are also written in the form of formula (1).[2]

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

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

Given the geometric objects $t_{i,1}, \ldots, t_{i,k_i}$, the function $\mathrm{O}i(t_{i,1},\ \ldots, t_{i,k_i})$ in (2) computes the fold line that satisfies operation (Oi), where i = 1, 2 and 4. The

[1] The most recent version of Eos requires an installation of Mathematica 9 or 10 and it can be downloaded from the webpage `http://www.i-eos.org/tutorial`.

[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.

equality "=" is polymorphic, and can be instantiated to be the equality between Lines, Points and Nums, as well. In formulas (3), O3, O5 and O6 are predicates, and not functions that return a fold line x since x may not be unique. For instance, $\mathrm{O5}(P, m, Q, x)$ states that there is a fold line x passing through point Q, and superposing point P and line m. There may be one or two fold lines, if exist.

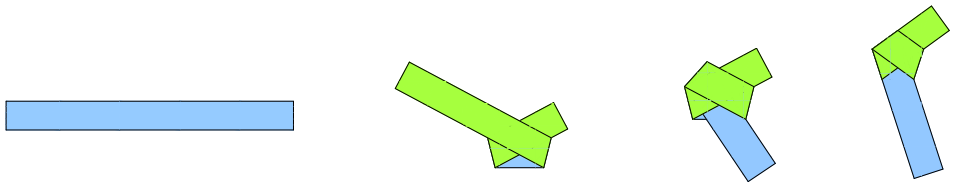
Function HO (which stands for Huzita Ori) allows the orgamist to interact with Eos and perform a fold operation specified by formula (1). Huzita's fold operations (O1) $\sim$ (O6) are implemented in Eos by translating them to formulas (2) and (3). The call HO(P, m, Q), for instance, asks Eos to internally treat the formula $\exists_{x,x:\mathrm{Line}}\ \mathrm{O5}(P, m, Q, x)$, and solve for x that satisfies $\mathrm{O5}(P, m, Q, x)$. The implementation of fold operation in Eos is extensible. The orgamist may ask the system to perform a fold operation that cannot be performed by Huzita's fold operations alone. The origamist can pass formula (1) to Eos as an argument of function HO.

$$\mathrm{HO}((\exists_{x_1,x_1:\tau_1} \ldots \exists_{x_i,x_i:\tau_i}\ \phi_1(t_{1,1}, \ldots, t_{1,k_1}) \wedge \ldots \wedge \phi_s(t_{s,1}, \ldots, t_{s,k_s})), \langle \textit{keyword arguments} \rangle)$$

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) $x_1, \ldots, x_i$, i.e. mountain or valley, and so on. Such information is given as optional arguments in HO call of the form "keyword $\rightarrow$ value". Otherwise, Eos undertakes these tasks using their default values.

3. Knot Fold of Regular Pentagon

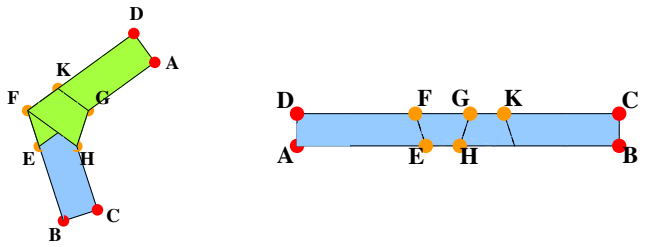
The construction of the simplest knot can be decomposed into the four steps shown in Fig. 1. We start with a rectangular origami or *origami tape* depicted in Fig. 1a. First, we perform two folds as shown in Fig. 1b. Next, we take the end of the upper face and mountain-fold it while inserting it immediately 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 constructing 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 variant of Alperin-Lang multi-fold operation [**1**]. The multi-fold, however, can be specified by properties of the knot fold given by a formula of the form (1). We first analyse the geometric properties of the knot. To this end,



(A) (B)

FIGURE 2. Unfolding knot fold of regular pentagon

we mark the key points of the knot, unfold it and examine the fold lines and the points that have been constructed during the knotting 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]

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

Steps 1 ∼ 3 are calls of EOS functions, i.e. the calls of BeginOrigami, NewPoint, and one HO. 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 ABCD of size 100×10. EOS defines a Cartesian coordinate system whose x-axis and y-axis are along lines AB and AD, respectively. Initial points A, B, C and D are located at (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 put the point E at (40, 0). At step 3, EOS, internally, computes the algebraic forms of the argument formula of HO, solves them, and 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.

We now explain the argument formula of HO. Referring to Fig. 2b, we establish the incidence relations between points and lines involved in the knot fold, i.e. $h \in \mathrm{AB} \wedge \{f,g,k\} \subset \mathrm{CD} \wedge f \in m \wedge h \in n \wedge k^n \in l$. 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 $\mathrm{O5}(g, \mathrm{EA}, \mathrm{E}, m) \wedge \mathrm{O5}(f, \mathrm{EB}, g, n) \wedge \mathrm{O5}(h, \overline{\mathrm{C}^n g}, f, l)$, we perform three (O5) operations. Regarding the third (O5), we first explain how to read Eos's notation $\overline{\mathrm{C}^n g}$ in $\mathrm{O5}(h, \overline{\mathrm{C}^n g}, f, l)$. Recall that $\overline{\mathrm{C}^n g}$ is the line passing through points C^n and g. The fold along line l passing through f superposes point h and line $\overline{\mathrm{C}^n g}$. The "over-bar" in line $\overline{\mathrm{C}^n g}$ is used here to correctly parse the line. 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. To make it a rigid knot, we need to pull the paper until all the points are moved to their "proper" locations in Fig. 2a. The difference immediately noticed 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 $\rightarrow$ 4" is added to HO to choose the solution that leads to a regular pentagon. The solutions for variables f, g, h and k are given the names F, G, H and K which is specified by argument "MarkPointAt $\rightarrow \{\mathrm{F}, \mathrm{G}, \mathrm{H}, \mathrm{K}\}$". By this way, the points bound to the existential variables become available in the following steps of the construction. The keyword argument "Handles $\rightarrow \{\mathrm{A}, \mathrm{B}, \mathrm{B}\}$" 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 $\rightarrow \{\mathrm{Valley}, \mathrm{Valley}, \mathrm{Mountain}\}$" asks Eos to perform valley folds along lines m and n and a mountain fold along line l. "InsertFace $\rightarrow \{0, 0, \mathrm{Bottom}\}$" 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. The outputs of the above HO call are shown in Fig. 4.

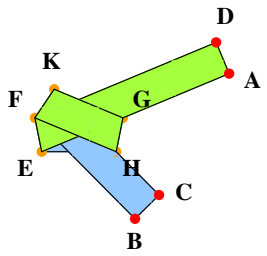
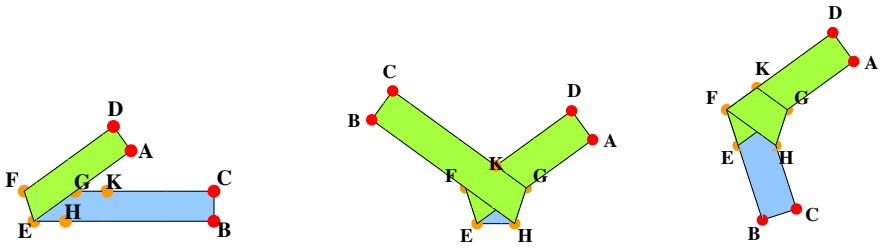


FIGURE 3. The knot fold before pulling the tape



(A) (B) (C)

FIGURE 4. Construction of regular pentagon EFKGH

4. Knot Fold of Regular Heptagon

We now examine the knot fold from an algebraic point of view. Through the example of the regular heptagon, we show another alternative formulation to define the constraints on knot fold using polynomial equations relating point locations. 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).

$$\begin{aligned}
&\exists_{m,m:\mathrm{Line}}\exists_{n,n:\mathrm{Line}}\\
&\exists_{f,f:\mathrm{Point}}\exists_{h,h:\mathrm{Point}}\exists_{g,g:\mathrm{Point}}\exists_{j,j:\mathrm{Point}}\exists_{k,k:\mathrm{Point}}\exists_{l,l:\mathrm{Point}}\\
&\exists_{ht,ht:\mathrm{Num}}\exists_{p,p:\mathrm{Num}}\exists_{q,q:\mathrm{Num}}\exists_{r,r:\mathrm{Num}}
\end{aligned}$$

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

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

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

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

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

The existentially quantified variables m and n of sort Line, f, g, h, j, k and l of sort Point, and ht, p and q of sort Num satisfy the constraints (4) $\sim$ (8). Relations (4) show the relations of incidences between points f, g, j, k and l and lines m, n, AB and CD. Equations (5) $\sim$ (7) state the following properties about the locations of points f, g, h, j, k and l with respect to the location of point E.

- The interior angles of the regular heptagon 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 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. $ht=|\mathrm{AD}|$, where $|\mathrm{AD}|$ denotes the distance between points A and D. 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 $X-Y$ for points X and Y is a coordinate-wise subtraction yielding a new point. Namely $\mathrm{Point}(x_1,\,y_1)-\mathrm{Point}(x_2,\,y_2)$ is $\mathrm{Point}(x_1-x_2,\,y_1-y_2)$. Thus, in expression (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 that $|\mathrm{EZ}|=|\mathrm{EF}|\times r$, and we deduce that $|\mathrm{EH}|=2ht\times q\times r$. The same property holds for the isosceles triangles $\triangle$JGK and $\triangle$KFJ, and hence we deduce the equalities in (6) and (7).

Regarding the polynomial equalities in (8), recall that $p=q\times\cos(3\theta)$. By the use of 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 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 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 on the half-line EA or on 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 regular heptagon ELHGJKF in Fig. 5.

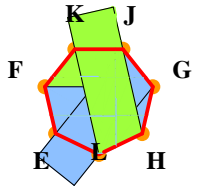


FIGURE 5. Knot fold of regular heptagon

5. Correctness of Knot Fold

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

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

Depending on the algebraic forms of $\mathcal{P}$ and $\mathcal{C}$, EOS first 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, EOS uses 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 employed by EOS Gröbner basis computation. Let $\mathcal{P}$ be $\forall \underline{x} \mathcal{P}'$ and $\mathcal{C}$ be $\forall \underline{x} \forall \underline{y} \mathcal{C}'$, where $\mathcal{P}'$ and $\mathcal{C}'$ are conjunctions of quantifier-free equalities, and the sequences of variables $\underline{x}$ and $\underline{y}$ are distinct. What we want to prove then is

$$\forall \underline{x} \forall \underline{y} (\mathcal{P}' \Rightarrow \mathcal{C}') \tag{9}$$

We prove proposition (9) by contradiction, i.e. to show that proposition (9) holds by showing the negative formula $\exists \underline{x} \exists \underline{y}(\mathcal{P}' \wedge \neg \mathcal{C}')$ does not hold. Algebraically, it is is reduced to finding the Gröbner basis of the set of polynomials generated from $\exists \underline{x} \exists \underline{y}(\mathcal{P}' \wedge \neg \mathcal{C}')$. Namely, if the reduced Gröbner basis is $\{1\}$, proposition (9) is true.

In the next two sections, we explain the proof of the correctness of the knot fold of regular pentagon and regular heptagon.

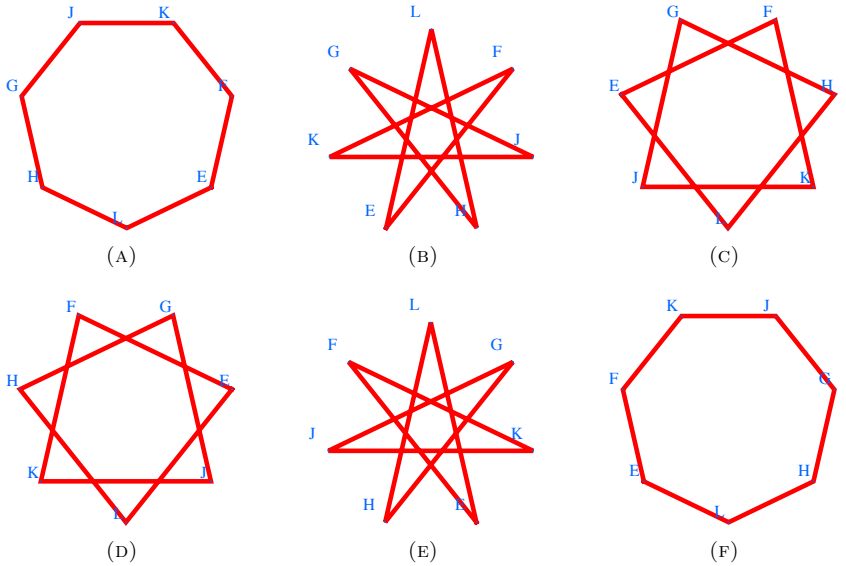


FIGURE 6. All the cases generated by function HO

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. Let O be the centre of EFKGH and $\theta = \measuredangle \mathrm{EOF}$. We use vectors to prove the equalities of the edges and of the incidence angles simultaneously. Namely, we prove that the vectors $\overrightarrow{\mathrm{FK}}$, $\overrightarrow{\mathrm{KG}}$, $\overrightarrow{\mathrm{GH}}$ and $\overrightarrow{\mathrm{HE}}$ are the rotations of $\overrightarrow{\mathrm{EF}}$, $\overrightarrow{\mathrm{FK}}$, $\overrightarrow{\mathrm{KG}}$ and $\overrightarrow{\mathrm{GH}}$, respectively, by angle θ around origin O, and, furthermore, that θ is equal to $\frac{\pi}{5}$. We use function Goal of Eos to specify these geometric properties.

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

Note that $\forall \underline{x}$ of proposition (9) is not specified since it can be generated by Eos automatically. Function $\mathrm{ToZ}(\overrightarrow{XY})$ computes the complex number $(v-u)+\imath\,(w-s)$ from points $X = \mathrm{Point}(u,\ s)$ and $Y = \mathrm{Point}(v,\ w)$. Hence, $\alpha\ \mathrm{ToZ}(\overrightarrow{XY})$ is the rotation of vector $\overrightarrow{XY}$ by an angle θ, where $\alpha = \cos(\theta) + \imath \sin(\theta)$.

We call function Prove to ask Eos to prove the correctness.

Prove("Regular knot pentagon",
Mapping → {A → {−w, 0}, B → {w, 0}, C → {w, 1}, D → {−w, 1}, E → {0, 0}}
Tactics → {Split,
Subgoal → {"Lemma Trapezoid",
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. The mapping attributes the coordinates (0, 0), (−w, 0), (w, 0), (w, 1) and (−w, 1) to points E, A, B, C, D, respectively. 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 edge AB of length $2w$.

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 lemma about the following equality. Folds in Fig. 1b are about making two congruent isosceles triangles △GFE and △FHG. Therefore, polygon EFGH in Fig. 1b is an isosceles trapezoid with |EF| = |HG| [**10**]. This equality of the segment lengths is expressed by formula "SquaredDistance(E, F) = SquaredDistance(H, G)". When introducing a subgoal $\mathcal{E}$ written as $\forall \underline{x} \forall \underline{z} \mathcal{E}'$, EOS shows that the two formulas $\forall \underline{x} \forall \underline{z} (\mathcal{P}' \Rightarrow \mathcal{E}')$ and $\forall \underline{x} \forall \underline{z} \forall \underline{y} (\mathcal{P}' \wedge \mathcal{E}' \Rightarrow \mathcal{C}')$. Note that in this particular case, the sequence $\underline{z}$ is empty. The introduction of the subgoal $\mathcal{E}$ is not necessary but has the advantage of considerably speeding up the computation of Gröbner basis.

The proofs of the above two formulas are successful. The CPU times used for computing Gröbner basis of the polynomials generated by the two formulas on Mac OS X (Intel Core i7 8GB 2.9GHz) are 42 seconds and 380 seconds, respectively.

5.3. Proof of Knot Fold of Regular Heptagon. Similarly to the proof of the knot fold of a regular pentagon, we specify a logical formula for the conclusion, and pass it to EOS.

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

We call function Prove to prove that the polygon EFKJGHL in Fig. 5 is a regular heptagon for arbitrary point E on edge AB.

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

Note that conclusion $\forall_{\alpha,\alpha:\mathbb{C}}\ \mathcal{C}'$ is of the form

$$\forall_{\alpha,\alpha:\mathbb{C}}(\mathcal{C}_1{}' \Rightarrow \mathcal{C}_2{}' \wedge \ldots \wedge \mathcal{C}_7{}'). \tag{10}$$

By writing "Tactics → {Split}", we ask EOS to split (10) into separate formulas $\forall_{\alpha,\alpha:\mathbb{C}} \mathcal{P}' \Rightarrow (\mathcal{C}_1{}' \Rightarrow \mathcal{C}_i{}')$, where $2 \leq i \leq 7$. EOS proves the six propositions independently. The proof is successful. The total time for computing Gröbner basis is 151 seconds on the same machine as the one used for the proof in Sect. 5.2.

6. Conclusion

We presented the construction of knot folds of a regular pentagon and a regular heptagon by specifying logically the geometric properties of the knots. The knotted origami is obtained by solving algebraic constraints generated from the logical specification. We further showed the proof of the correctness of the construction. The examples presented in this paper are available in the webpage `http://www.i-eos.org/tutorial`.

With the increase of the number of the edges of the knotted polygons, the degree of the algebraic equations as well as the number of equations and variables increases. This requires further computation time of Gröbner basis, and to reduce the computation time remains a challenging tasks. Towards proving knot fold of regular $2(n \geq 9)+1$-gons in a reasonable computation time, we need to investigate strategies for optimization, e.g. orderings of the variables.

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. H. Huzita, *Axiomatic Development of Origami Geometry*, Proceedings of the First International Meeting of Origami Science and Technology, 1989, pp. 143–158.
6. 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.
7. T. Ida and H. Takahashi, *Origami Fold as Algebraic Graph Rewriting*, Journal of Symbolic Computation **45** (2010), 393–413.
8. T. Ida, H. Takahashi, M. Marin, A. Kasem, and F. Ghourabi, *Computational Origami System Eos*, Origami^4, Proceedings of the Fourth International Meeting of Origami Science, Mathematics, and Education (4OSME), 2009, pp. 285–293.
9. J. Maekawa, *Introduction of the Study of Knot Tape*, Origami^5, Proceedings of the Fourth International Meeting of Origami Science, Mathematics, and Education (5OSME), 2011, pp. 395–403.
10. 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`