

Morley's Theorem Revisited: Origami Construction and Automated Proof

Tetsuo Ida, Asem Kasem, Fadoua Ghourabi,

Department of Computer Science, University of Tsukuba, Tsukuba 305-8573, Japan

Hidekazu Takahashi

Shiga Prefectural Yokaichi High School, Higashiomi, 527-0022, Japan

Abstract

Morley's theorem states that for any triangle, the intersections of its adjacent angle trisectors form an equilateral triangle. The construction of Morley's triangle by the straightedge and compass is impossible because of the well-known impossibility result of the angle trisection. However, by origami, the construction of an angle trisector is possible, and hence of Morley's triangle. In this paper we present a computational origami construction of Morley's triangle and automated correctness proof of the generalized Morley's theorem.

During the computational origami construction, geometrical constraints in symbolic representation are generated and accumulated. Those constraints are then transformed into algebraic forms, i.e. a set of polynomials, which in turn are used to prove the correctness of the construction. The automated proof is based on the Gröbner bases method. The timings of the experiments of the Gröbner bases computations for our proofs are given. They vary greatly depending on the origami construction methods, algorithms for Gröbner bases computation, and variable orderings.

Key words: Morely's Theorem, Computational Origami, Automated Geometrical Theorem Proving, Gröbner Bases

* This research is supported by the JSPS Grants-in-Aid for Exploratory Research No. 19650001 and Scientific Research (B) No. 20300001.

Email addresses: ida@score.cs.tsukuba.ac.jp (Tetsuo Ida), kasem@score.cs.tsukuba.ac.jp (Asem Kasem), ghourabi@score.cs.tsukuba.ac.jp (Fadoua Ghourabi), t.hidekazu@gmail.com (Hidekazu Takahashi).

1. Introduction

Computational origami is a scientific discipline to study mathematical and computational aspects of origami. It includes the mathematical study of paper folds, modeling of origami by algebraic and symbolic methods, computer simulation of paper folding, and proving the correctness of geometrical properties of constructed origami. In the framework of computational origami we studied the construction of Morley's triangles and automated proofs of Morley's theorem. Morley's theorem states that the three points of intersection of the adjacent trisectors of the angles of any triangle form an equilateral triangle. Morley's theorem can be generalized by taking into account the intersections of the trisectors of the exterior angles as well. For a given angle α ($0 < \alpha < \pi$), we have one pair of interior angle trisectors producing the pair of angles $(\alpha/3, 2\alpha/3)$, and the other two pairs of exterior angle trisectors producing the pairs of angles $((\pi+2\alpha)/3, (2\pi+\alpha)/3)$ and $((2\pi+2\alpha)/3, (4\pi+\alpha)/3)$. Therefore, we have 3^3 possible triangles formed by the intersections of the adjacent angle trisectors. The generalized Morley's theorem states that out of the 27 triangles constructible by the intersections of the adjacent angle trisectors, 18 triangles are equilateral. Proofs of (generalized) Morley's theorem were published by several researchers since Morley gave his result in 1898. Bogomolny gives a comprehensive account on Morley's theorem in his web page (Bogomolny, 1996).

In this paper, we present a computational origami construction of Morley's triangles and prove automatically the correctness of the generalized Morley's theorem in a streamlined fashion as observed and coined as proving-computing-solving style by Buchberger (2004). The process is realized by the computational origami system called EOS(E-origami system) Ida et al. (2006). The automated proof of the generalized Morley's theorem was first published by Wu (1986) using Wu-Ritt method. A concise explanation of his proof is given in Wang (2004). The computational origami construction and the streamlined automated proof are new in our study. Seemingly different kinds of knowledge in mathematical sciences, i.e. origami and automated theorem proving, are integrated in a common framework and, moreover, processed coherently. Origami represented as a set of equalities is systematically transformed into a polynomial set. The generated set of polynomials is input to Gröbner bases computation algorithms, and the proof is completed.

The rest of the paper is organized as follows. After briefly explaining the principles of origami construction in section 2, in section 3 we show a stepwise origami construction of a Morley's triangle. In section 4 we give the automated proof of the generalized Morley's theorem. We presents the experimental results of computing Gröbner bases in section 5. In section 6 we summarize our results and indicate directions for future research.

This paper is the revised version of the paper Ida et al. (2005) presented at the Fourth Symposium on Mathematical Knowledge Management 2005, and incorporates the progress of the research since then. The extended abstract of this paper was presented at Application of Computer Algebra, 2008, Session on Gröbner Bases and their Applications.

2. Principles of Origami Construction

2.1. Origami Foldability

An origami¹ is folded along a line on the origami called *fold line*. The fold line is specified by the origamist. In subsection 2.2 we recall the six basic origami axioms proposed

¹ Origami is a Japanese word meaning a sheet of folding (ori) paper (gami) or methodology of folding a paper.

by Huzita (1989) to fold an origami. Each of his axioms prescribes a rule for constructing a fold line, which can be determined by either points, lines and/or combinations of them. It is known that Huzita's origami axiom set is more powerful than the straightedge and compass method in Euclidean plane geometry (Geretschl ager, 2002). Namely, origami can construct some geometrical objects that are impossible to construct by the straightedge and compass method. One of them is a trisector of an arbitrary angle. The impossibility of the construction by the straightedge and compass was shown by Wantzel (1837). It is one of the three famous impossibilities (See, for example, the textbook by Jones et al. (1994)). Hence, Morley's triangles can not be constructed by the straightedge and compass method.

Now let us see how to construct a geometrical object with an origami. First, we have some notational convention in this paper. We denote points by single capital letters A, B, \dots possibly subscripted. Expression XY can be either the line passing through points X and Y or the segment between points X and Y . The distinction can be made easily by the context. We define an origami² $\square ABCD$ together with the set Π of constructible points $\{A, B, C, D\}$ and the set Γ of constructible lines $\{AB, BC, CD, DA\}$. The constructible points and lines of origami are defined by Alperin (2000). We then start the origami construction from the initial origami $\square ABCD$. We make a fold on the origami by applying one of the axioms given below, possibly followed by unfolding. A fold of the origami gives rise to a set of new points of intersection of the fold line and the lines in Γ , resulting in new Π and Γ .

2.2. Huzita's axioms

In Huzita (1989), Huzita proposed the following axiom set $\{(O1), \dots, (O6)\}$ for origami geometry. Let Π and Γ be the set of constructible points and of constructible lines, respectively.

- (O1) Given two points in Π , we can make a fold along the fold line that passes through them.
- (O2) Given two points in Π , we can make a fold to bring one of the points onto the other.
- (O3) Given two lines in Γ , we can make a fold to superpose the two lines.
- (O4) Given a point P in Π and a line m in Γ , we can make a fold along the fold line that is perpendicular to m and passes through P .
- (O5) Given two points P and Q in Π and a line m in Γ , either we can construct the fold line that passes through Q and make a fold along this fold line to superpose P and m , or we can decide that the construction of such a fold line is impossible.
- (O6) Given two points P and Q in Π and two lines m and n in Γ , either we can construct a fold line and make a fold along this fold line to superpose P and m , and Q and n , simultaneously, or we can decide that the construction of such a fold line is impossible.

Later, an additional axiom was proposed by Hatori (2005):

- (O7) Given a point P in Π and two lines m and n in Γ , either we can construct the fold line that is perpendicular to n and make a fold along this fold line to superpose P and m , or we can decide that the construction of such a fold line is impossible.

² We abuse the word *origami* to mean the methodology, the sheet of paper, and the geometrical object that is being constructed by means of paper folds, as we do in Japanese.

He further showed that (O6) is sufficient to make all the folds by (O1) – (O5) and (O7). Indeed, (O1) – (O5) and (O7) are the degenerate cases of (O6). This does not mean, however, that (O6) is enough in practice. As we see shortly in the application of (O6), (O6) will deliver at most three fold lines. We would need to specify an additional parameter to select the desired fold line.

Mathematical models of the set of constructible points by the folds by the applications of axiom sets are studied later and independently by Alperin (2000).

2.3. Implementation of Huzita's axioms

As we are interested in computational aspects of origami, and further in turning origami into modern engineering technology and a pedagogical methodology for geometry, we are naturally led to consider the implementation of origami based on the Huzita's axioms. The implementation of (O1) – (O7) can be systematically derived as described by Ida et al. (2007). For instance, (O5) is about the statement

$$\begin{aligned} \forall P, Q \in \Pi \forall m \in \Gamma \exists \text{ a fold line } l \text{ such that} \\ (Q \text{ is on line } l) \wedge (\text{reflection of } P \text{ with respect to } l \text{ is on line } m) \end{aligned} \quad (1)$$

The formula (1) is true if there exists such a fold line l . To see the existence we solve the algebraic constraint for l .

The solver for Huzita's axioms is implemented as function `HFold` in EOS. As EOS is implemented in *Mathematica* we use the *Mathematica* notation for functional representation. As shown below, function `HFold` needs, as arguments, several constructible lines and points to compute the fold line(s) and to determine the origami face to be moved.

- (O1) `HFold[X, Along \rightarrow PQ]`
- (O2) `HFold[P, Q]`
- (O3) `HFold[RS, UV]`
- (O4) `HFold[X, AlongPerpendicular \rightarrow {P, RS}]`
- (O5) `HFold[P, RS, Through \rightarrow Q]`
- (O6) `HFold[P, RS, Q, UV]`
- (O7) `HFold[P, RS, AlongPerpendicular \rightarrow UV]`

Note that the types of the arguments and the argument keywords can discriminate the operations to be performed unambiguously for each axiom. `HFold[X, Along \rightarrow PQ]` in (O1) makes a fold along the line extending the segment PQ . The words `Along`, `AlongPerpendicular` and `Through` to the left of \rightarrow are the keywords of the parameter specification. Point X specifies the side of the fold line, from which we determine the faces that should be moved. In all the cases we have omitted optional parameters which tell `HFold` which faces of the origami should be moved (with keyword `Move`) and which directions (with keyword `Mountain` or `Valley`). For instance, in (O2), `Move \rightarrow P`, `Direction \rightarrow Valley` is implicit. The figures that we will show in the next section are generated by the calls of `HFold`.

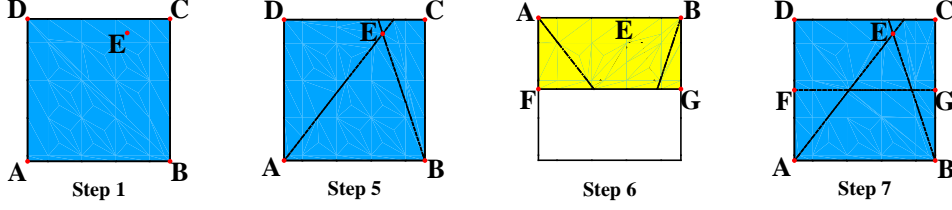


Fig. 1. Origami construction of a Morley's triangle: Steps 1 – 7, steps 2 – 4 are omitted

3. Origami Construction of a Morley's Triangle

3.1. Preparation

The origami construction of a Morley's triangle will be shown in Figs. 1 – 9. In the initial origami $\square ABCD$, we put an arbitrary point E (which we assume constructible), and construct a Morley's triangle inside the triangle $\triangle ABE$ (cf. Fig. 1, step 5).

We will perform trisections of $\angle EAB$ (steps 6 – 13), of $\angle ABE$ (steps 14 – 19) and of $\angle BEA$ (steps 20 – 29). Then we see the triangle $\triangle LRS$ as shown in Fig. 9. The points L , R , and S are the intersections of the two adjacent trisectors of the angles of $\triangle ABE$.

In this paper we show two methods of the construction of angle trisectors; one by Abe's method described in Fushimi (1980) and the other by the multifold method of Alperin and Lang (2009).

3.2. Abe's method

To trisect the angle $\angle EAB$, we need a perpendicular to AB at point A ; the line AD will do, and a line parallel to AB and equidistant from points D and A . The parallel is obtained by applying (O2) to bring point A to D , and then unfold (steps 6 and 7 in Fig. 1). The line FG is the desired parallel. The names of new points are automatically generated by EOS, unless we specify them as parameters of `HFold`.

Step 8 is the crucial step of Abe's method, which involves application of (O6). We make a fold to bring point D and point A onto line AE and line FG , respectively. Finding a fold line in (O6) amounts to solving a cubic equality that describes the geometrical constraints among the involved points and lines. In the straightedge and compass method, all constructible numbers are algebraic over the field of rational numbers \mathbb{Q} and have degree of power of two over \mathbb{Q} . Thus construction of fold lines that are determined by the roots of cubic equality is impossible using the straightedge and compass method. However, Huzita's axiom (O6) does solve the cubic equality. Since the system solves the cubic equality, we have (at most) three possible fold lines that satisfy (O6) as shown in Fig. 2.

At this step, we need to interact with EOS to specify which fold line we want to use. In our example we will choose the one in case 3 in the figure. The fold line in case 1 is used to trisect the angle $(\pi - \angle EAB)$, and the fold line in case 2 is used to trisect angle $(2\pi - \angle EAB)$.

At step 8 in Fig. 3, we make the fold along the fold line of case 3, and then we project the points A and F by `PointProject[A, F]`. When we unfold the origami at step 9, we see the new points H and I , the projections of points A and F , respectively, appear. Lines AH and AI are the trisectors of $\angle EAB$.

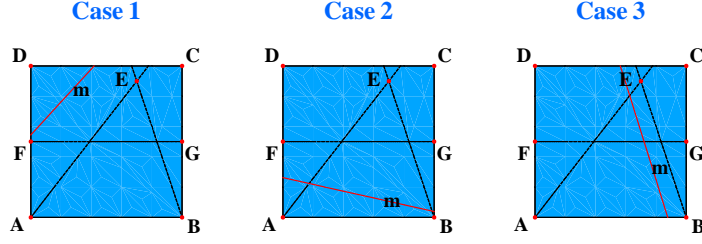


Fig. 2. Three possible fold lines in the application of (O6): $\text{HFold}[D, AE, A, FG]$

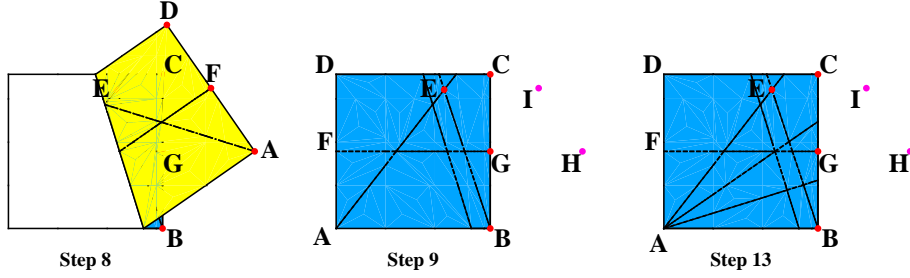


Fig. 3. Origami construction of a Morley's triangle: Steps 8 – 13, steps 10 – 12 are omitted

As we already constructed the parallel to AB , trisecting the $\angle ABE$ is done easily by applying (O6), to superpose point C and point B onto line BE and line FG , respectively. We choose the fold line of case 1 (Fig. 4) from the possible fold lines, and make a fold and an unfold (steps 14 and 15, respectively, in Fig. 5). The steps 16 – 19 are simply for the folds to create the trisectors through the marked points.

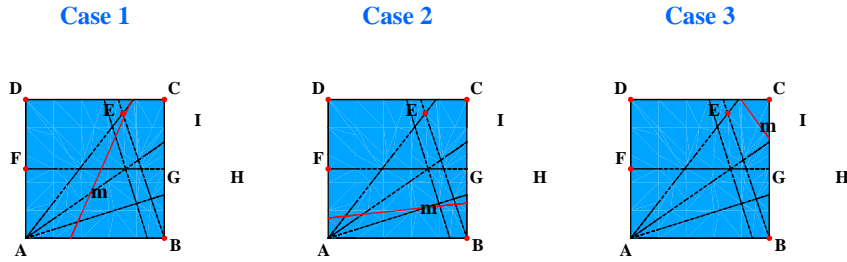


Fig. 4. Three possible fold lines in the application of (O6): $\text{HFold}[C, BE, B, FG]$

Finding the trisector of $\angle BEA$ is more involved (steps 20 – 29 in Figs. 6, 7, and 8). To construct the parallel to BE , we first construct a perpendicular to BE at point E using (O4), i.e. by the call of $\text{HFold}[D, \text{AlongPerpendicular} \rightarrow \{E, BE\}]$. The point M is the intersection of the fold line and AD . To obtain the parallel, we make a fold using (O2), to superpose points E and M (steps 20 – 23). The rest of the construction to obtain the trisectors is similar to those of angles $\angle EAB$ and $\angle ABE$.

The final origami is shown in Fig. 9. We see the triangle $\triangle LRS$, which we will prove to be equilateral in section 4.

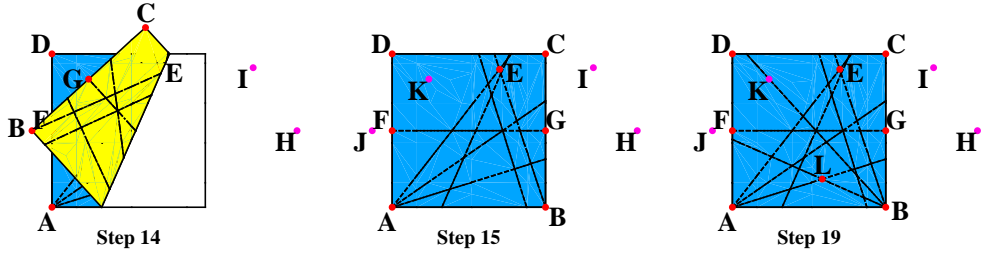


Fig. 5. Origami construction of a Morley's triangle: Steps 14 – 19, steps 16 – 18 are omitted

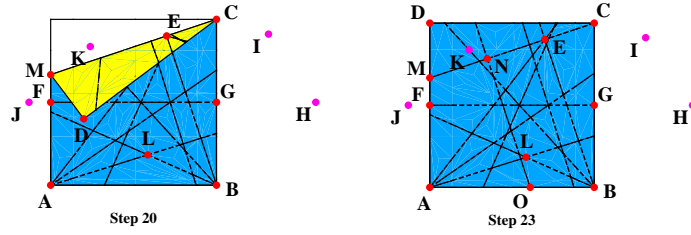


Fig. 6. Construction steps 20 – 23, steps 21,22 are omitted

Case 1

Case 2

Case 3

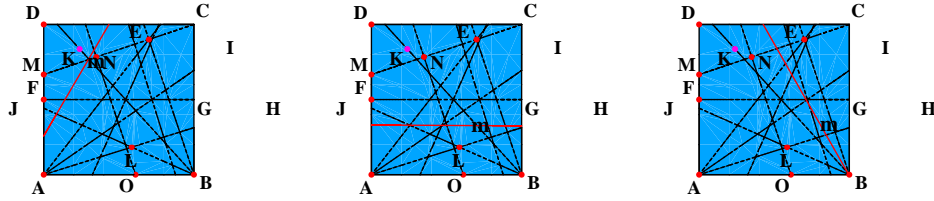


Fig. 7. Three possible fold lines in the application of (O6): $\text{HFold}[M, AE, E, ON]$

3.3. Multifold Method

Origamists agree that all the constructions pertaining to Huzita's axioms can be performed by hands. The foldability by hands in the case of axioms (O1) – (O4) is obvious. Regarding to axioms (O5) – (O7), origamits have to slide a given point along a given line to make the constructions possible. We accept the argument for this kind of sliding a point by hands. Another fold method, called *multifold method*, which would extend the notion of foldability by hands is proposed by Alperin and Lang (2009). It would allow an origamist to make folds along multiple fold lines. Double folds can be performed by hands without losing much of the precision.

However, more than "2-fold" folding is difficult. Although whether a human origamist can make multiple folds is debatable, in this paper we incorporate the multifold as another basic fold operation extending Huzita's axioms.

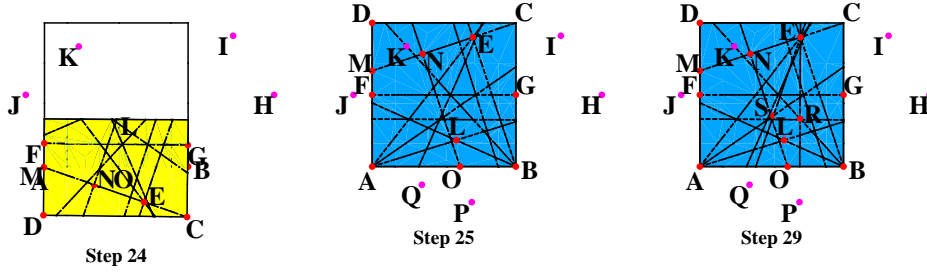


Fig. 8. Origami construction of a Morley's triangle: Steps 24 – 29, steps 26 – 28 are omitted

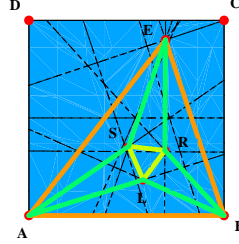


Fig. 9. Morley's Triangle: Final Step

3.3.1. Multifold by EOS

In EOS, the extension to the multifold is natural as `HFold` is implemented with the generality that allows the incorporation of the multifold. Multifold is realized by the call of the following:

$$\text{HFold} [\mathcal{H}, \mathcal{L}]$$

\mathcal{H} is a list of points on origami which determine the faces to be moved. \mathcal{L} is a formula in the first-order predicate logic. The formula specifies the constraints that the geometrical objects concerned have to satisfy. In the case of the multifold, \mathcal{L} specifies the constraints that the fold lines should satisfy. All Huzita's axioms are implemented as an instance of the general `HFold` function.

To trisect an angle, we perform a double-fold operation which is a simultaneous application of two (O5). Given an angle $\angle EAB$, we need to find lines m and n that are trisectors passing through A . The reflection of B with respect to m is on n , and the reflection of E with respect to n is on m . Therefore, m and n are the fold lines that bring B onto n and E onto m simultaneously.

To trisect $\angle EAB$ on the origami in Fig. 1, we make the following call of `HFold`

$$\begin{aligned} &\text{HFold}[\{B, E\}, \exists_{\{m, n\}, \{m \in \text{Line}, n \in \text{Line}\}} \\ &\quad (\text{OnLine}[\text{Reflection}[B, m], n] \wedge \text{OnLine}[A, m] \wedge \\ &\quad \text{OnLine}[\text{Reflection}[E, n], m] \wedge \text{OnLine}[A, n])] \end{aligned} \quad (2)$$

The atomic formula `OnLine` $[P, m]$ states that the point P is on the line m . The term `Reflection` $[P, m]$ represents the reflection of P with respect to m . Thus, the formula `OnLine` $[\text{Reflection}[P, n], m] \wedge \text{OnLine}[Q, n]$ states that the reflection of P with respect to n is on m and n passes through Q , which is the formula corresponding to (1) in subsection 2.3.

The result of the evaluation of (2) is shown in Fig. 10. There are three cases that satisfy the formula, as in the case of Abe's method. In case 2, m and n are trisectors of $\angle EAB$, whereas in cases 1 and 3, m and n are trisectors of angles $(2\pi - \angle EAB)$ and $(\pi - \angle EAB)$, respectively. We choose case 2 since in the construction of Morley's triangle we want to trisect the internal angles of $\triangle ABE$. After unfolding, we obtain the desired trisectors as shown in Fig. 11.

We use the multifold method to simplify the steps of trisecting angles. The construction of the vertices L , R and S of the Morley's triangle proceeds in the same way described in Abe's method.

3.3.2. Algebraic Interpretation

This subsection is devoted to the algebraic interpretation of Huzita's axioms. This algebraic interpretation is used to "evaluate" the function calls of `HFold` as well as the automated proof of the correctness of the construction. We take `HFold` in (2) as an example.

To obtain m and n , the formula in (2) is transformed into a set of algebraic equalities. An atomic formula is interpreted as a set of polynomial equalities, and a term is given as a rational function. For an atomic formula ϕ , $\llbracket \phi \rrbracket$ denotes the set of polynomial equalities that are the algebraic meaning of ϕ . Let ϕ and ψ be two atomic formulas, we define

- $\llbracket \phi \wedge \psi \rrbracket = \llbracket \phi \rrbracket \cup \llbracket \psi \rrbracket$
- $\llbracket \phi \vee \psi \rrbracket = \{p \mid q = 0 \mid (p = 0) \in \llbracket \phi \rrbracket, (q = 0) \in \llbracket \psi \rrbracket\}$
- $\llbracket \neg \phi \rrbracket = \{\prod_{(p=0) \in \llbracket \phi \rrbracket} (p \xi_p - 1) = 0\}$, where ξ_p is a slack variable introduced by Rabinowitch trick.

The method for the algebraic interpretation is detailed in (Ghourabi et al., 2007). The set of (non-simplified) polynomial equalities (3) – (8) is the algebraic interpretation of the formula in the `HFold` call of (2).

$$\{b4(0.a3^2 + 0.b3^2 - 2b3(1.a3 + c3)) + a4(-1.a3^2 + 1.b3^2 - 2a3(0.b3 + c3)) + c4(a3^2 + b3^2) = 0, \quad (3)$$

$$0.a3 + 0.b3 + c3 = 0, \quad (4)$$

$$b3(0.9a4^2 - 0.9b4^2 - 2b4(0.7a4 + c4)) + a3(-0.7a4^2 + 0.7b4^2 - 2a4(0.9b4 + c4)) + c3(a4^2 + b4^2) = 0, \quad (5)$$

$$0.a4 + 0.b4 + c4 = 0, \quad (6)$$

$$(-1 + b4)b4 = 0, (-1 + a4)(-1 + b4) = 0, -1 + (1 + a4^2)\kappa3 = 0, \quad (7)$$

$$(-1 + b3)b3 = 0, (-1 + a3)(-1 + b3) = 0, -1 + (1 + a3^2)\kappa4 = 0 \quad (8)$$

Note that we work in a Cartesian coordinate system where $A(0,0)$, $B(1,0)$, $E(0.7,0.9)$, which we see in Fig. 1. A line $a x + b y + c = 0$ is represented by (a, b, c) , together with the constraint $(-1 + b)b = 0 \wedge (-1 + a)(-1 + b) = 0 \wedge a^2 + 1 \neq 0$.

Now, we examine the equalities (3) and (4) that are the algebraic interpretation of the sub-formula `OnLine[Reflection[B, m], n] \wedge OnLine[A, m]`. Let R be the reflection of B on n with respect to m . The coordinates of R are

$$\left(\frac{-1.a3^2 + 1.b3^2 - 2a3(0.b3 + c3)}{a3^2 + b3^2}, \frac{0.a3^2 + 0.b3^2 - 2b3(1.a3 + c3)}{a3^2 + b3^2} \right).$$

The algebraic interpretation of $\text{OnLine}[\mathbf{R}, \mathbf{n}]$ is

$$\frac{b4(0.a3^2 + 0.b3^2 - 2b3(1.a3 + c3))}{a3^2 + b3^2} + \frac{a4(-1.a3^2 + 1.b3^2 - 2a3(0.b3 + c3))}{a3^2 + b3^2} + c4 = 0 \quad (9)$$

To obtain the polynomial equality (3), we canceled the denominators by multiplying the both sides of (9) by $a3^2 + b3^2$. The relation $a3^2 + b3^2 \neq 0$ is ensured by (8). Equality (4) states that A is on line m . In the same way, the equalities (5) and (6) are derived from $\text{OnLine}[\text{Reflection}[\mathbf{E}, \mathbf{n}], \mathbf{m}] \wedge \text{OnLine}[\mathbf{A}, \mathbf{n}]$. By solving the above set of polynomial equalities for the coefficients of m and n , we obtain the fold lines m and n .

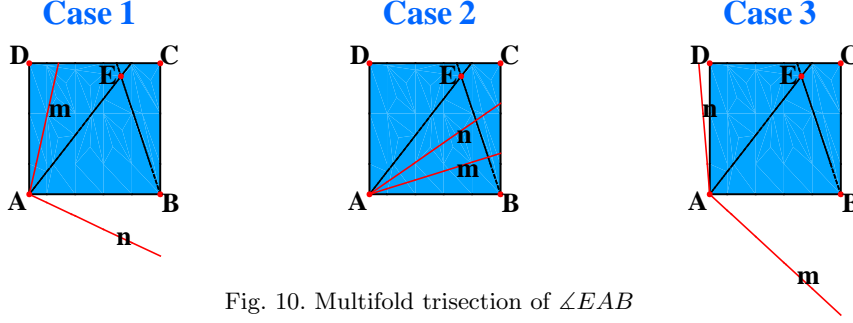


Fig. 10. Multifold trisection of $\angle EAB$

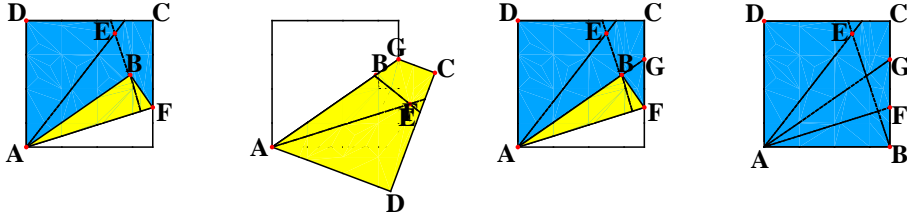


Fig. 11. Multifold followed by unfold

4. Proof of Morley's Theorem

4.1. Algebraic formulation of Morley's theorem

The essence of the construction is finding fold lines, i.e. solving the geometrical constraints for the fold lines. The geometrical constraints that have been accumulated during the construction will be used for proving too. We would like to remark that construction and proving can be interleaving. In this section, we explain the correctness proof for the construction by Abe's method for the generalized Morley's theorem. What has to be proven is that

- (i) $\triangle LRS$ (cf. Fig. 9) is an equilateral triangle.
- (ii) The fold lines constructed at steps 13, 19 and 29 are trisectors.

The automated proofs of (ii) can be made immediately after the construction steps 13, 19 and 29. We omitted those proofs here since the proof technique is the same as what

we will expound in due course. Moreover, in Ida and Buchberger (2003), it is shown that Abe's method constructs trisectors using Gröbner bases method.

So we proceed to prove (i). We will show that after the construction, the following holds:

$$d(L, R)^2 = d(R, S)^2 \wedge d(L, R)^2 = d(S, L)^2, \quad (10)$$

where $d(X, Y)$ denotes the distance between points X and Y . Let \mathcal{K} be the geometrical constraints accumulated during the construction, and \mathcal{C} be the formula (10). \mathcal{K} and \mathcal{C} form the premise and conclusion of the proposition (11) that we want to prove:

$$\mathcal{K} \Rightarrow \mathcal{C}. \quad (11)$$

However, the specification of the premise requires careful analysis. The construction that we have shown is one particular instance of the construction, and the geometrical constraints that are accumulated during the construction are general ones that admit all the possible constructions, namely the constructions of all 27 triangles. To see this point let us recall that during the construction, EOS allowed us to choose a fold line when other two fold lines are possible (steps 8, 14, and 24). This choice is necessary to proceed with the construction and visualize the ensuing origami. However, the choice is not reflected on the geometrical constraints accumulated during the construction. Therefore, the solution of the obtained polynomials contains all possible fold lines, and thus all 27 possible triangles.

We know that 9 triangles out of the constructed 27 triangles are not equilateral. Figure A.1 shows all the triangles. Wu (1986) articulated the situation. We have to single out the 18 equilateral triangles, by imposing the following condition on the angles. Let α , β and γ be angles $\angle LAB$, $\angle ABL$, and $\angle BER$, respectively.

$$\alpha + \beta + \gamma = \pm\pi/3 \pmod{2\pi}$$

which implies the following:

$$\tan^2(\alpha + \beta + \gamma) = 3$$

We let $t_1 = \tan \alpha$, $t_2 = \tan \beta$, $t_3 = \tan \gamma$ and by straightforward trigonometric manipulations we obtain the following condition in polynomial equality:

$$\mathcal{X} : (t_1 + t_2 + t_3 - t_1 t_2 t_3)^2 - 3(1 - t_1 t_2 - t_2 t_3 - t_3 t_1)^2 = 0$$

We revise the proposition (11) of the generalized Morley's theorem to the following formula:

$$\mathcal{L} : \mathcal{K} \wedge \mathcal{X} \Rightarrow \mathcal{C} \quad (12)$$

Let $\mathcal{E}_{\mathcal{L}}$ be the set of polynomial equalities representing formula \mathcal{L} , $\mathcal{P}_{\mathcal{L}}$ be the set of the polynomials $\{p \mid p = 0 \in \mathcal{E}_{\mathcal{L}}\}$, and $\text{Ideal}(\mathcal{S})$ be the ideal generated by set \mathcal{S} of polynomials. Formula \mathcal{L} is true if

$$1 \in \text{Ideal}(\mathcal{P}_{-\mathcal{L}}). \quad (13)$$

The ideal membership problem $1 \in \text{Ideal}(\mathcal{S})$ can be solved constructively by computing the Gröbner bases of \mathcal{S} . Namely, the statement (13) is true iff the reduced Gröbner basis of $\mathcal{P}_{-\mathcal{L}}$ is $\{1\}$. Hence, \mathcal{L} is true if the reduced Gröbner basis of $\mathcal{P}_{-\mathcal{L}}$ is $\{1\}$. $\mathcal{P}_{-\mathcal{L}}$ is obtained in the following way. We first note that $\neg \mathcal{L}$ is logically equivalent to $\mathcal{K} \wedge \mathcal{X} \wedge \neg \mathcal{C}$. Then each constraint is transformed to the algebraic equalities as discussed in subsection 3.3.2.

4.2. Proof by EOS

We are now ready to prove the generalized Morley's theorem. The premise \mathcal{K} of the theorem are the geometrical constraints accumulated during the construction. We first give to Eos the auxiliary condition \mathcal{X} by calling function **AssertProp**:

```
AssertProp[  $\exists_{\{t1,t2,t3\},\{t1 \in \text{Alg}, t2 \in \text{Alg}, t3 \in \text{Alg}\}}$ 
   $t1 == \text{ToTangent}[L, A, B] \wedge$ 
   $t2 == \text{ToTangent}[A, B, L] \wedge$ 
   $t3 == \text{ToTangent}[B, E, R] \wedge$ 
   $(t1 + t2 + t3 - t1 t2 t3)^2 == 3(1 - t1 t2 - t1 t3 - t2 t3)^2]$ 
```

where **ToTangent** $[X, Y, Z]$ gives $\tan \angle XYZ$ as a rational function of its point coordinates. Now, $\mathcal{K} \wedge \mathcal{X}$ forms the revised premise of the Morley's theorem.

We then add the conclusion by calling the function **Goal**:

```
Goal[Distance[L, R]2 == Distance[R, S]2  $\wedge$  Distance[L, R]2 == Distance[L, S]2]
```

where **Goal** $[formula]$ adds the negation of the conclusion to the premise to obtain $\mathcal{K} \wedge \mathcal{X} \wedge \neg \mathcal{C}$.

In order to translate $\mathcal{K} \wedge \mathcal{X} \wedge \neg \mathcal{C}$ into algebraic form, we fix the coordinate system to be Cartesian with points A, B, C, D and L as follows:

```
cmap = Mapping[{ {A, Point[0, 0]}, {B, Point[1, 0]}, {C, Point[1, 1]}, {D, Point[0, 1]},
  {L, Point[u1, u2]} ]
```

Without loss of generality, we set the size of the initial origami to be 1×1 . The point L is taken to be arbitrary. One may wonder why point E , instead of L , was not taken to be arbitrary. This is because of the efficiency of the Gröbner bases computation. Wang (2008) observed that by letting L to be arbitrary but fixed, the size of the search space during the Gröbner bases computation is greatly reduced. When L is taken to be arbitrary, the two trisectors are determined, whereas if E is taken in that way, there will be no restrictions on the choice of trisectors.

Finally, we check whether the reduced Gröbner basis of the algebraic interpretation of $\mathcal{X} \wedge \mathcal{K} \wedge \neg \mathcal{C}$ is $\{1\}$ by calling **Prove**.

```
Prove[Mapping  $\rightarrow$  cmap
  CoefficientDomain  $\rightarrow$  RationalFunctions,
  MonomialOrder  $\rightarrow$  DegreeReverseLexicographic]
```

Function **Prove** performs the following operations:

- 1 Selecting the relevant constraints for proving the generalized Morley's theorem: Based on the predicates specified in the argument of function **Goal**, the program selects the geometrical properties that are necessary for the proof. In the case of the generalized Morley's theorem, the geometrical properties related to points L, R and S are selected. Recall that points L, R and S form the Morley's triangle in Fig. 9.
- 2 Generating the algebraic interpretation of $\mathcal{K} \wedge \mathcal{X} \wedge \neg \mathcal{C}$: The geometrical constraints are transformed into the set \mathcal{S} of polynomials.
- 3 Ordering the variables: Based on the order of construction steps, the program computes the ordered list \mathcal{V} of variables in \mathcal{S} .
- 4 Computing Gröbner basis of \mathcal{S} : The Gröbner basis computation is carried out in the domain of polynomials whose variables are in $\mathcal{V} \setminus \{u1, u2\}$ and whose coefficients are in $\mathbb{Q}(u1, u2)$ of rational functions.

5. Experiments

Geometrical constraints in origami may grow exponentially in the worst case due to the very nature of folding. Furthermore, each Huzita’s axiom generates at least 5 equalities; constructed points may be moved by reflections; and constraints are generated for finding intersections. Taking all those factors into account, one can see easily that the numbers of the generated polynomials and of the introduced variables grow rapidly. This significantly affects the time required to compute Gröbner bases.

Therefore, we usually need several trial runs to compute Gröbner bases by various monomial orders, variable orders, and computing algorithms. Although the Gröbner bases computation is guaranteed to terminate theoretically, it is also probable that computing programs may abort due to lack of memory.

In the case of the construction of a Morley’s triangle, the number of generated polynomials by Abe’s method was 292 having 190 variables, and by the multifold method was 135 having 94 variables. We investigated the problem of finding a proper variables order based on the order of creation of variables, and by the type of generated polynomial equalities in which they are involved. In our experiments, we ordered the variables according to the construction step in which they were generated. Therefore, we have the variables used for points coordinates and line coefficients \succ variables introduced for forming auxiliary conditions and the conclusion.

With the above remarks in mind, we will summarize the result of our experiments. We used *Mathematica7* to compute Gröbner bases on a machine equipped with Intel Core 2 Duo 2.67 GHz processor, and 2.00 GB of memory. *Mathematica* provides two algorithms for the computation, **Buchberger** (Buchberger, 1985) and **GroebnerWalk** (Collart et al., 1997) algorithms. We tried with lexicographic, degree lexicographic, and degree reverse lexicographic monomial orders. We obtained the results shown in Table 1. No entry in the table signifies the failure of the computation due to lack of memory. The timings are in seconds, where we obtained {1} as the result, indicating that the theorem was proved.

Table 1. Computation times by *Mathematica7* - Construction by Abe’s method

	Lexicographic	Degree Lexicographic	Degree Reverse Lexicographic
Buchberger			867
GroebnerWalk	881	857	877

In the case of using multifold method for origami construction, we computed Gröbner bases using the same hardware and obtained the results shown in Table 2. They show a significant improvement to prove the theorem using the multifold method, due to the reduced number of generated variables (and polynomials). We also note that changing the algorithms or monomial orders had small effect on the time required for computing the bases except for the case where we used **Buchberger** algorithm.

Experiments results can be found at <http://www2.score.cs.tsukuba.ac.jp/eos/apps>.

Table 2. Computation times by *Mathematica7* - Construction by multifold method

	Lexicographic	Degree Lexicographic	Degree Reverse Lexicographic
Buchberger	29.2	6.6	8.8
GroebnerWalk	9.0	9.1	8.9

6. Conclusion

We have shown the origami construction of a Morley's triangle and the automated proof of the generalized Morley's theorem using Gröbner bases method. Using the computational origami system EOS, we not only perform the origami construction with rigor and ease exceeding those by paper fold by hands, but also prove the correctness of the construction automatically. We have observed that solving (i.e. origami construction) and proving are interleaving and interactive. The computing time for the origami construction using EOS is not a problem. Rather we benefit very much from the capabilities of symbolic algebra and advanced graphics of *Mathematica*. However, the automated theorem proving part of our work required large amount of time and efforts to bring the results into the present form. We needed many trials to choose the right monomial orders. Without the detailed analyses of the generated polynomials and the optimizations of geometrical constraints, Gröbner bases computations would not be successfully completed. We therefore see that the development of computing environment such as web services to facilitate the trials of Gröbner bases computations is a challenging research to pursue.

References

- Alperin, R. C., 2000. A Mathematical Theory of Origami Constructions and Numbers. New York Journal of Mathematics 6, 119–133.
- Alperin, R. C., Lang, R., 2009. One-, two-, and multi-fold origami axioms. In: Lang, R. (Ed.), Fourth International Meeting of Origami Science, Mathematics and Education (4OSME), 2006. A K Peters Ltd, pp. 371 – 393.
- Bogomolny, A., 1996. Morley's Miracle.
URL <http://www.cut-the-knot.org/triangle/Morley>
- Buchberger, B., 1985. Groebner-Bases: An Algorithmic Method in Polynomial Ideal Theory. In: Bose, N. (Ed.), Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems. Copyright: Reidel Publishing Company, Dordrecht - Boston - Lancaster, The Netherlands, Ch. 6, pp. 184–232, (Second edition: N.K.Bose (ed.): Multidimensional Systems Theory and Application, Kluwer Academic Publisher, 2003, pp.89-128.).
- Buchberger, B., 2004. Algorithm Supported Mathematical Theory Exploration: A Personal View and Strategy. In: Campbell, J. (Ed.), Proceedings of 7th International Conference on Artificial Intelligence and Symbolic Computation (AISC 2004). Vol. 3249 of Lecture Notes in Artificial Intelligence. Springer, pp. 236–250.
- Collart, S., Kalkbrener, M., Mall, D., 1997. Converting Bases with the Gröbner Walk. Journal of Symbolic Computation 24 (3-4), 465–469.
- Fushimi, K., October 1980. Science of Origami. A supplement to Saiensu (in Japanese).

- Geretschläger, R., 2002. Geometric Constructions in Origami. Morikita Publishing Co., in Japanese, translation by Hidetoshi Fukagawa.
- Ghourabi, F., Ida, T., Takahashi, H., Marin, M., Kasem, A., 2007. Logical and Algebraic View of Huzita's Origami Axioms with Applications to Computational Origami. In: Proceedings of the 22nd ACM Symposium on Applied Computing. ACM Press, pp. 767–772.
- Hatori, K., 2005. K's Origami - Fractional Library - Origami Construction.
URL <http://origami.ousaan.com/library/conste.html>
- Huzita, H., 1989. Axiomatic Development of Origami Geometry. In: Proceedings of the First International Meeting of Origami Science and Technology. pp. 143–158.
- Ida, T., Buchberger, B., 2003. Proving and Solving in Computational Origami. In: Analele Universitatii de Vest din Timisoara. Vol. XLI, Fasc. Special of Mathematica-Informatica. pp. 247–263.
- Ida, T., Marin, M., Takahashi, H., Ghourabi, F., 2007. Computational Origami Construction as Constraint Solving and Rewriting. In: 16th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2007). Electronic Notes in Theoretical Computer Science. R. Echahed, pp. 139–152.
- Ida, T., Takahashi, H., Țepeneu, D., Marin, M., 2005. Morley's Theorem Revisited through Computational Origami. In: Proceedings of the 7th International Mathematica Symposium (IMS 2005).
- Ida, T., Takahashi, H., Marin, M., Ghourabi, F., Kasem, A., 2006. Computational Construction of a Maximal Equilateral Triangle Inscribed in an Origami. In: Mathematical Software (ICMS 2006). Vol. 4151 of Lecture Notes in Computer Science. Springer, pp. 361–372.
- Jones, A., Morris, S. A., Pearson, K. R. (Eds.), 1994. Abstract Algebra and Famous Impossibilities, 2nd Edition. Springer-Verlag.
- Wang, D., 2004. Elimination Practice: Software Tools and Applications. Imperial College Press London.
- Wang, D., 2008. Personal Communication.
- Wantzel, M., 1837. Recherches sur les moyens de reconnaître si un Problème de Géométrie peut se résoudre avec la règle et le compas. Journal de Mathématiques Pures et Appliquées 1 (2), 366–372.
- Wu, W., 1986. Basic Principles of Mechanical Theorem Proving in Elementary Geometry. Journal of Automated Reasoning 2, 221–252.

A. Triangles Generated by All Trisectors

The triangles drawn in the bold lines are given triangles, and those in the thinner lines (red in color) are the triangles formed by the intersections of adjacent trisectors. Although the sizes of the given triangles are varied, they are similar. Eighteen of the constructed triangles are equilateral.

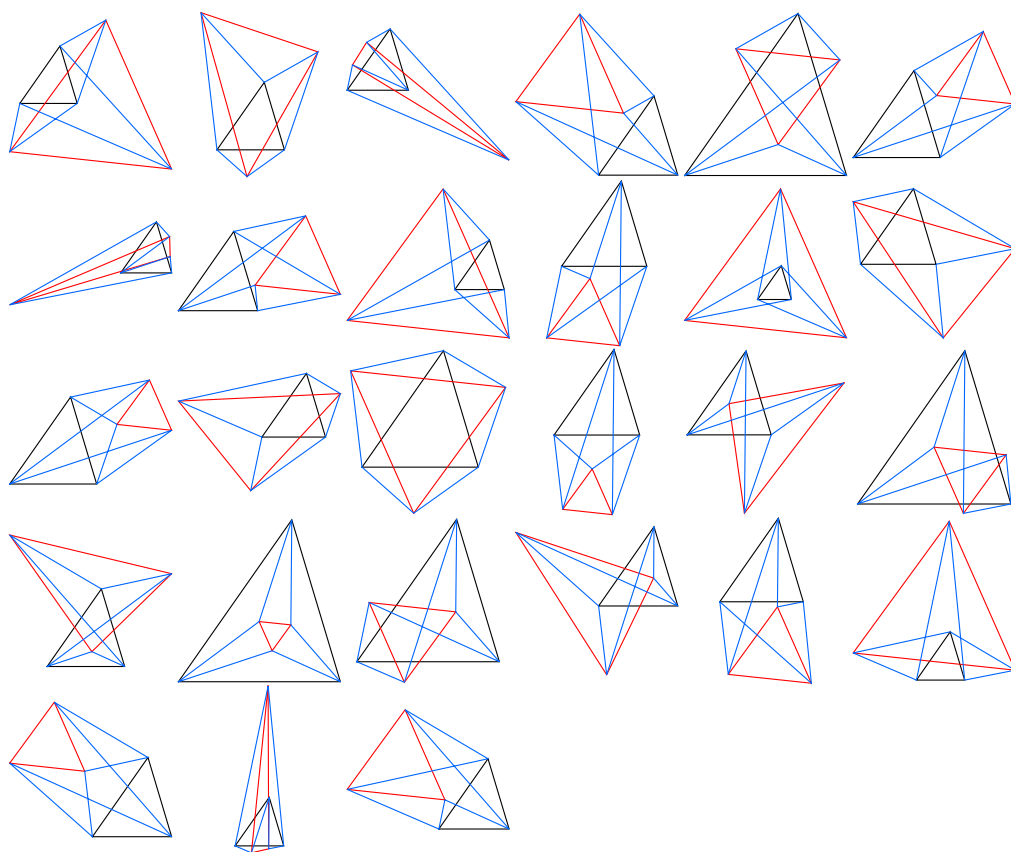


Fig. A.1. The 27 possible triangles