Analele Universităţii din Timişoara
Vol. XLI, Fasc. special, 2003
Seria Matematică–Informatică

PROVING AND SOLVING IN COMPUTATIONAL ORIGAMI

Presented at the 5th International Workshop on Symbolic and Numeric Algorithms for Scientific Computation

Tetsuo IDA, Bruno BUCHBERGER

Abstract. Origami (paper folding) has a long tradition in Japan's culture and education. We are developing a computational origami system, based on symbolic computation system Mathematica, for performing and reasoning about origami on the computer. This system is based on the implementation of the six fundamental origami folding steps (origami axioms) formulated by Huzita. In this paper, we show how our system performs origami folds by constraint solving, visualizes each step of origami construction, and automatically proves general theorems on the result of origami construction using algebraic methods. We illustrate this by a simple example of trisecting an angle by origami. The trisection of an angle is known to be impossible by means of a ruler and a compass. The entire process of computational origami shows nontrivial combination of symbolic constraint solving, theorem proving and graphical processing.

Key words: Origami, symbolic computation, theorem proving, Theorema

AMS Subject Classification: 68T15

1. Introduction Origami is a Japanese traditional art of paper folding. The word origami is a combined word coming from ori (fold) and kami (paper). For several centuries, origami has been popular among Japanese common people as an art, as a playing toy, and teaching material for children. Over a decade, origami is also receiving wide interest among mathematicians, mathematics educators and computer scientists, as well as origami artists, as origami poses interesting fundamental geometrical questions.

We are proposing computational origami, as part of a discipline of origami science (also coined origamics by Haga, a pioneer in Japanese origami research [Haga 1999]). We believe that the rigor of paper folding and the beauty of origami artworks enhance greatly when the paper folding is supported by a computer. In the earlier work of the first author [Ida 2003], computational origami performs paper folding by solving both symbolically and numerically certain geometrical constraints, followed by the visualization of origami by computer graphics tools. In this paper we extend the system for proving the correctness of the origami constructions, as proposed in our previous paper [Buchberger 2003].

Origami is easy to practice. Origami is made even easier with a computer: we can construct an origami by calling a sequence of origami a folding function on the Mathematica Notebook [Wolfram 2004] or by the interaction with our system, running in the computing server, using the standard web browser. The constructed final result, as well as the results of the intermediate steps, can be visualized and manipulated. Moreover, in cases where a proof is needed, the system will produce the proof of the correctness of the construction. Namely, for a given sequence of origami construction steps and a given property, the system proves that the resulting shape will satisfy the property (or disprove the property).

The rest of this paper is organized as follows. In section 2 we explain the principles of origami construction developed from Huzita's origami axioms. In section 3 we discuss a method for trisecting an angle by origami, which cannot be made using the traditional ruler-and-compass method. The construction is followed by the proof of the correctness of the construction in section 4. In section 5 we summarize our contributions and point out some directions for future research.

2. Principles of Origami Construction An origami is to be folded along a specified line on the origami called *fold line*. The line segment of a fold line on the origami is called a *crease*, since the consecutive operations of a fold and an unfold along the same fold line makes a crease on the origami. A fold line can be determined by the points it passes through or by the points (and/or lines) it brings together. As in Euclidean geometry, by specifying points, lines and their configuration, we have the following six basic fold operations called *origami axioms* of Huzita [Huzita 1989, Hull 1997]. It is known that Huzita's origami axiom set is more powerful than the ruler-and-compass method in Euclidean geometry [Geretschlaeger 2002]. Origami can construct objects that are impossible by the ruler-and-compass method [Chen 1966]. One of them is trisecting an angle, which we will show in this paper, as an example of origami proving and solving.

2.1. Origami Axioms Huzita's origami axioms are described in term of fold operations as follows:

(O1) Given two points, we can make a fold along the crease passing through them.

(O2) Given two points, we can make a fold to bring one of the points onto the other.

(O3) Given two lines, we can make a fold to superpose the two lines.

(O4) Given a point P and a line m, we can make a fold along the crease that is perpendicular to m and passing through P.

(O5) Given two points P and Q and a line m, either we can make a fold along the crease that passes through Q, such that the fold superposes P and m, or we can determine that the fold is impossible.

(O6) Given two points P and Q and two lines m and n, either we can make a fold along the crease, such that the fold superposes P and m, and Q and n, simultaneously, or we can determine that the fold is impossible.

Algorithmically, these axioms suggest two operations: finding crease(s) and folding the origami along the crease. The former amounts to solving equations that describe the constraints of geometrical objects. It can be shown easily that finding the creases is reduced to solving geometrical constraints (maximum 3rd degree polynomial system).

2.2. OFold (Origami Fold Function) The fold operations of origami based on the origami axioms can be implemented by a single function OFold whose specification is given below. Function OFold needs several points (depending on the axioms) to compute the crease, to determine the face to be moved, and to determine the direction of the fold (mountain or valley). The convention below is that a single capital letter denotes a point and XY denotes a line segment from point X to point Y. Note that the types of the arguments and the keywords can discriminate the operations to be performed unambiguously.

(O1) `OFold[X, Along → PQ]`

(O2) `OFold[P, Q]`

(O3) `OFold[AB, CD]`

(O4) `OFold[X, AlongPerpendicular → {P, AB}]`

(O5) `OFold[P, AB,Through → Q]`

(O6) `OFold[P, AB, Q, CD]`

`OFold[X, Along → PQ]` in (O1) directs the system to make a fold along the crease PQ. All the faces containing the values (the coordinate of the point X) are to be moved. In all the case we have hidden optional parameters which tell the system which faces of the origami should be moved (with `Move` keyword) and which directions (mountain or valley). In the case of (O2) , `Move → P` is implicit, and in the cases of (O5) and (O6) `Move → P` is implicit if with P we can specify the faces to be moved. Further details are omitted here, as the above specification is sufficient to understand the origami construction in section 3.

3. Trisecting an angle We give an example of trisecting an angle. This example shows a non-trivial use of Axiom (O6). The method of construction is due to H. Abe as described in [Geretschlger 2002, Fushimi 1980]. In the following we will explain the construction of trisecting a given angle using our computational origami system.
Step 1: First, we define a square origami paper, whose corners are designated by the points A, B, C and D. The size may be arbitrary, but for our example, let us fix it to be 10 by 10. All the operations are performed by Mathematica function calls. Optional parameters can be specified by "keyword → value".

```
NewOrigami[Square[10, MarkPoints → {"A", "B", "C", "D"}],
FigureCaption → "Step "];
```

(*See Fig.1*)
Step 2: We then introduce an arbitrary point, say E at the coordinate $(2, 10)$, assuming that point A is at $(0, 0)$.

```
PutPoint[{"E", Point[2, 10]}];
```

(*See Fig.1*)
Our problem is to trisect an angle $\angle EAB$. The method consists of the following seven steps (steps 3-9) of folds and unfolds.
Step 3: We make a fold to bring point A to D, to obtain the perpendicular bisector of segment ADa This is the application of (O2). The points F and G, which will be the terminal points of the crease, are automatically generated by the system.

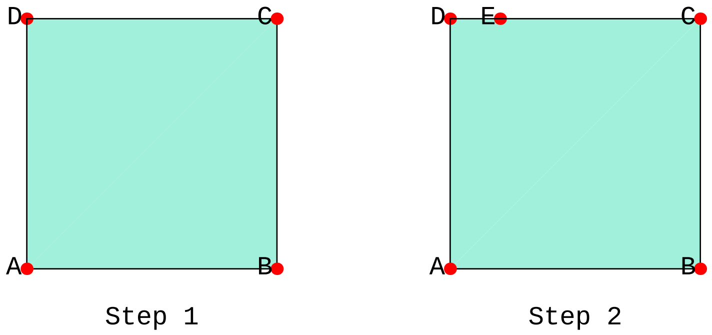

Fig. 1. Results of Steps 1 and 2

```
OFold[A, D];
```

(*See Fig.2*)

Step 4: We unfold the origami and obtain the crease FG

```
Unfold[];
```

(*See Fig.2*)

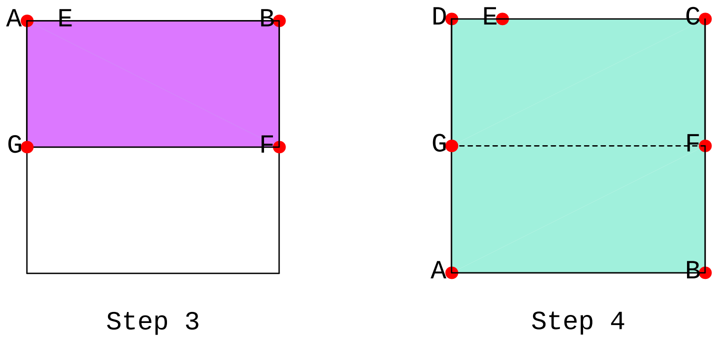

Fig. 2. Results of Steps 3 and 4

Steps 5 and 6: Likewise we obtain the crease HI.

```
OFold[A, G];
Unfold[];
```

(*See Fig. 3*)

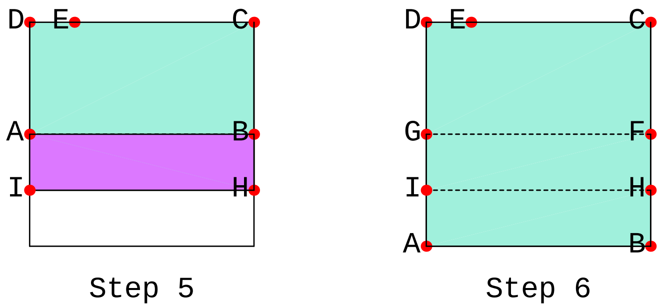

Fig. 3. Results of Steps 5 and 6

Step 7: This step is the crucial step of the construction. We apply (O6). We try to superpose G and the line that is the extension of the segments AE, and to superpose A and the line that is the extension of the segment HI, simultaneously. There are three possible fold lines to realize this superposition. The system responds with the query of "Specify the line number" together with the lines on the origami image.

```
OFold[G, AE, A, HI];
```

Which line(1,2,3)?

(*See Fig.4*)

Step 8: We reply to the query by the call of `OFold[Along→ 3,Move→A, ...]`, which tells the system that we choose the line number 3. This gives the fold line that we are primarily interested in. However, readers can easily see that the other two fold lines are also solutions (which trisect different angles).

```
OFold[Along → 3, Move → A, MarkCrease → False];
```

(*See Fig.4*)

Step 9: We will copy the points A and I on the other face that is below the face that A and I are on, and unfold the origami. The copied points appear as L and J (the names are automatically generated).

Specify the line number.

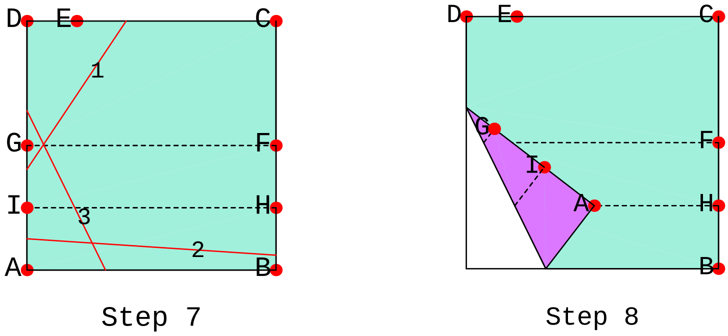

Fig. 4. Results of Steps 7 and 8

```
CopyPoint[{A, I}]; Unfold[];
```

(*See Fig.5*)

Step 10: Now we see (Fig.5) that the segments AJ and AL trisect the angle $\angle EAB$.

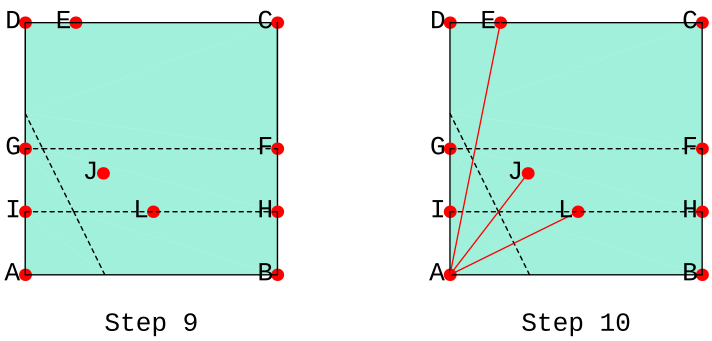

Fig. 5. Results of Step 9 and 10

Although it is not obvious to see that equational solving is performed, our system solves a set of polynomial equations, up to the third degree, at each step. In the case of steps 7 and 8, the system solves a cubic equation. This explains why we have (at most) 3 possible fold lines at step 7.

4. Proof of the correctness of the trisection method We now prove the following theorem with our system.

Theorem 1. *The origami construction in section 3 trisects an angle.*

In this simple example, the correctness of the trisection could be easily verified either by geometric reasoning or by a sequence of simplification steps of the algebraic equations representing geometric constrains. However, for proceeding towards a general (and completely automatic) proving method for origami theorems, we formulate the proving steps in a more general setting. We prove the above theorem by showing that

$$\measuredangle EAB/3 = \measuredangle JAB/2 = \measuredangle LAB.$$

A general proof procedure is as follows:

I We first translate the above question into the algebraic form. This is done after we fix the coordinate system (in our case the Cartesian system).

II We already observed that each folding steps are formulated in Axioms (O1)-(O6). The geometric properties that hold for each origami axiom are easily extracted in terms of polynomial constraints, once the representations of lines and points are fixed.

III We use Gröbner bases method. We collect all the premise equalities $C = \{c_1, \ldots, c_n\}$ (obtained at step II) and the conclusion equalities $D = \{d_1, \ldots, d_m\}$ (obtained at step I) . Let M be the boolean combinations of the equalities of the form $\neg(c_1 \wedge \ldots \wedge c_n) \vee (d_1 \wedge \ldots \wedge d_m)$, i.e. $C \Rightarrow D$. We prove $\forall M$ by refutation. The decision algorithm, roughly, proceeds as follows:

- Bring M into conjunctive normal form and distribute $\forall$ over the conjunctive parts. Treat each of the parts

$$\forall_{a,b,c,\ldots} P$$

separately. Note that P is now a disjunction

$$E_1 = 0 \vee \ldots \vee E_k = 0 \vee N_1 \neq 0 \vee \ldots \vee N_l \neq 0$$

of equalities and negations of equalities.

- Then

$$\underset{a,b,c,\ldots}{\forall}(E_1 = 0 \vee \ldots \vee E_k = 0 \vee N_1 \neq 0 \vee \ldots \vee N_l \neq 0)$$

is transformed into

$$\neg \underset{a,b,c,\ldots}{\exists}(E_1 \neq 0 \wedge \ldots \wedge E_k \neq 0 \wedge N_1 = 0 \wedge \ldots \wedge N_l = 0)$$

and further into

$$\begin{aligned}\neg \underset{a,b,c,\ldots,\xi_1,\ldots,\xi_k}{\exists} & (E_1\xi_1 - 1 = 0 \wedge \ldots \wedge E_k\xi_k - 1 = 0 \wedge \\ & N_1 = 0 \wedge \ldots \wedge N_l = 0)\end{aligned}$$

with new variables $\xi_1, \ldots, \xi_k$ ("Rabinovich trick").

- Now, our question becomes a question on the solvability of a system of polynomial equations, which can be decided by computing the reduced Gröbner basis of $\{E_1\xi_1 - 1, \ldots, E_k\xi_k - 1, N_1, \ldots, N_l\}$. Namely, one of the fundamental theorems of Gröbner bases theory tells us that this Gröbner basis will be $\{1\}$ iff the system is unsolvable (i.e. has no common zeros) [Buchberger 1970].

The proof steps would require laborious work of symbolic manipulation. We have developed a system for finding geometrical constraints generated in each step of the origami fold and for performing the above transformations completely automatically.
The following piece of programs will do all the necessary transformations and the proof as mentioned above.

```
ShowProofSupport[]
```

(*See Fig.6*)

The subscript k on each point indicates that the positions are determined at origami folding step k.

4.1. Preparing the Proof We first gather necessary geometric constraints that will become the premise of the theorem to be proved.

```
props = GatherProperty[];
```

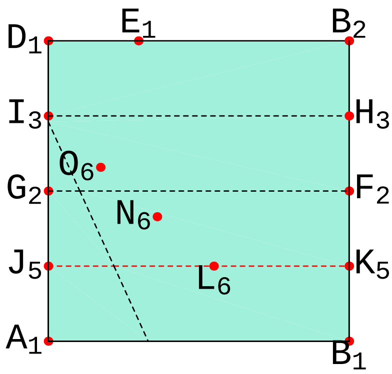

Fig. 6. Result of `ShowProofSupport[]`

We then fix the coordinate system to be Cartesian by calling function CoordinateMapping, which return a mapping table `cmap`.

```
cmap  = CoordinateMapping[props,InitialShape ->
        SquareP[{Point[0, 0], r}, {"A", "B","C", "D"}]];
```

The premise is generated from the geometric constraints stored in the variable props using the coordinate mapping table cmap.

```
premise = ToAlgebraic[props, cmap];
```

Let us now turn to defining the conclusion polynomials. Let $\gamma = \angle EAB$, $\beta = \angle JAB$ and $\alpha = \angle KAB$. Our conclusion is $\beta = 2\alpha \wedge \gamma = 3\alpha$. Since function tangent restricted to $(0, \pi/2)$ is bijective, we will prove $\tan[\beta] = \tan[2\alpha] \wedge \tan[\gamma] = \tan[3\alpha]$. We compute them by:

$$\begin{aligned} \tan\gamma &= E_1/.\texttt{Point}[x\_, y\_, \_\_\_] :\to y/x \\ \tan\beta &= J_6/.\texttt{Point}[x\_, y\_, \_\_\_] :\to y/x \\ \tan\alpha &= L_6/.\texttt{Point}[x\_, y\_, \_\_\_] :\to y/x \end{aligned}$$

Using the well-known elementary trigonometric formulas, we have:

$$\tan 3\alpha = \texttt{Simplify}\left[\frac{3\texttt{Tan}[x] - \texttt{Tan}[x]^3}{1 - 3\texttt{Tan}[x]^2}/.\texttt{Tan}[x] \to \tan\alpha\right]$$

$$\tan 2\alpha = \texttt{Simplify}\left[\frac{2\texttt{Tan}[x]}{1 - \texttt{Tan}[x]^2}/.\texttt{Tan}[x] \to \tan\alpha\right]$$

The conclusion polynomial set is {Numerator[Together[tanγ-tan3α]]=0, Numerator[Together[tanβ-tan2α]]=0}, which can be translated to algebraic form as follows. The variables were automatically generated when the coordinate mapping table was constructed.

$$\text{concl} = \texttt{ToAlgebraic[\{ Numerator[Together[}\tan\beta - \tan 2\alpha\texttt{]],}$$
$$\texttt{Numerator[Together[}\tan\gamma - \tan 3\alpha\texttt{]]\}]}$$

$$\{ -3x10^2\ x3\ y10 + x3\ y10^3 + x10^3\ y3 - 3\ x10\ y10^2\ y3,$$
$$-2\ x10\ x9\ y10 + x10^2\ y9 - y10^2\ y9\}$$

4.2. Sending all the data to theorem prover Theorema We have also developed a simple interface to Theorema which is running on a remote server (conceptually, it can be in the same computer as the computational origami system is running).

We first create the link with Theorema with the port 50000 at the machine whose IP address is 192.168.11.2 (in this illustration). Appropriate arrangements on the remote server side running Theorema are necessary. This arrangements are not shown in this paper.

```
thma = LinkCreate[  "50000@192.168.11.2",
                    LinkProtocol → "TCPIP" ];
```

We then send data stored in the variables `premise` and `concl` via the link `thma` and wait for the proof to come from Theorema.

```
SendTheoremaFormula[  thma, premise, concl,
                      "TrisectingAngle"];
```

Once the proof text arrived, we save it in the file TrisectingAngleProof.nb.

```
NotebookSave[LinkRead[thma], "TrisectingAngleProof.nb"];
```

Finally we close the link.

```
LinkClose[thma];
```

4.3. The Proof The proof text is stored in file named TrisectingAngleProof.nb. Since it takes too much space to reproduce the machine generated (albeit readable) text output, we only show the highlights of the proof. The omitted formulas are marked as "...".
Prove:

(Formula (TrisectingAngle): (1))

$$\forall_{\substack{a5,a6,a7,a8,b5,b6,b7,b8,c5,c6,c7,c8,r,x1,x10,x2,x3,x4,x5,\ x6,x7,\\ x8,x9,y1,y10,y2,y3,y4,y5,y6,y7,y8,y9,\eta1,\eta2,\eta3,\eta4,\eta5,\mu1,\mu2,\mu4}}$$

$((c8 = 0) \wedge (a5*r = 0) \wedge (c5 + \frac{1}{2}*b5*r = 0) \wedge (-1*r + x4 = 0) \wedge (x5 = 0) \wedge$
$(-1*r + x7 = 0) \wedge (x8 = 0) \wedge (b5*(r + (-1)*x1) + a5*y1 = 0) \wedge$
$(c5 + \frac{1}{2}*a5*(r + x1) + \frac{1}{2}*b5*y1 = 0) \wedge (-1*b7*x10 + a7*y10 = 0) \wedge$
$(c7 + \frac{1}{2}*a7*x10 + \frac{1}{2}*b7*y10 = 0) \wedge (b6*(r + (-1)*x2) + a6*y2 = 0) \wedge$
$(c6 + \frac{1}{2}*a6*(r + x2) + \frac{1}{2}*b6*y2 = 0) \wedge (c8 + a8*x3 + b8*y3 = 0) \wedge$
$(c5 + a5*x4 + b5*y4 = 0) \wedge (-1*b6*x5 + a6*y5 = 0) \wedge$
$(c5 + a5*x5 + b5*y5 = 0) \wedge (c6 + \frac{1}{2}*a6*x5 + \frac{1}{2}*b6*y5 = 0) \wedge$
$(b7*(x5 + (-1)*x6) + a7*(-1*y5 + y6) = 0) \wedge$
$(c7 + \frac{1}{2}*a7*(x5 + x6) + \frac{1}{2}*b7*(y5 + y6) = 0) \wedge (c6 + a6*x7 + b6*y7 = 0) \wedge$
$(c6 + a6*x8 + b6*y8 = 0) \wedge (b7*(x8 + (-1)*x9) + a7*(-1*y8 + y9) = 0) \wedge$
$(c7 + \frac{1}{2}*a7*(x8 + x9) + \frac{1}{2}*b7*(y8 + y9) = 0) \wedge (-1 + r*\eta1 = 0) \wedge$
$(-1 + (a5^2 + b5^2)*\eta2 = 0) \wedge (-1 + (a6^2 + b6^2)*\eta3 = 0) \wedge$
$(-1 + (a7^2 + b7^2)*\eta4 = 0) \wedge (-1 + (a8^2 + b8^2)*\eta5 = 0) \wedge$
$(c8 + a8*\mu1 + b8*\mu2 = 0) \wedge (b7*(x5 + (-1)*\mu1) + a7(-1*y5 + \mu2) = 0) \wedge$
$(c7 + \frac{1}{2}*a7*(x5 + \mu1) + \frac{1}{2}*b7*(y5 + \mu2) = 0) \wedge (-1*b7*\mu3 + a7*\mu4 = 0) \wedge$
$(c6 + a6*\mu3 + b6*\mu4 = 0) \wedge (c7 + \frac{1}{2}*a7*\mu3 + \frac{1}{2}*b7*\mu4 = 0) \Rightarrow$
$(-3*x10^2*x3*y10 + x3*y10^3*y3 + (-3)*x10*y10^2*y3 = 0) \wedge$
$(-2*x10*x9*y10 + x10^2*y9 + (-1)*y10^2*y9 = 0))$
with no assumptions.

Proved.
The Theorem is proved by the Groebner Bases method.

The formula in the scope of the universal quantifier is transformed into an equivalent formula that is a conjunction of disjunctions of equalities and negated equalities. The universal quantifier can then be distributed over the individual parts of the conjunction. By this, we obtain:

Independent proof problems:

(Formula (TrisectingAngle): (1).1)

$$\forall_{\substack{a5,a6,a7,a8,b5,b6,b7,b8,c5,c6,c7,c8,r,x1,x10,x2,x3,x4,x5,\ x6,x7,\\ x8,x9,y1,y10,y2,y3,y4,y5,y6,y7,y8,y9,\eta1,\eta2,\eta3,\eta4,\eta5,\mu1,\mu2,\mu4}}$$

$$\begin{aligned}
&((-2*x10*x9*y10+x10^2*y9+(-y10^2*y9)=0)\vee c8\neq 0\vee x5\neq 0\vee x8\neq 0\vee\\
&-1+r*\eta1\neq 0\vee -1+a5^2*\eta2+b5^2*\eta2\neq 0\vee -1+a6^2*\eta3+b6^2*\eta3\neq 0\vee\\
&-1+a7^2*\eta4+b7^2*\eta4\neq 0\vee -1+a8^2*\eta5+b8^2*\eta5\neq 0\vee c5+\frac{1}{2}*b5*r\neq 0\vee\\
&(-r)+x4\neq 0\vee(-r)+x7\neq 0\vee b5*r+(-b5*x1)+a5*y1\neq 0\vee\\
&b6*r+(-b6*x2)+a6*y2\neq 0\vee b7*x5+(-b7*x6)+(-a7*y5)+a7*y6\neq 0\vee\\
&b7*x5+(-a7*y5)+(-b7*\mu1)+a7*\mu2\neq 0\vee\\
&b7*x8+(-b7*x9)+(-a7*y8)+a7*y9\neq 0\vee(-b6*x5)+a6*y5\neq 0\vee\\
&(-b7*x10)+a7*y10\neq 0\vee(-b7*\mu3)+a7*\mu4\neq 0\vee c5+a5*x4+b5*y4\neq 0\vee\\
&c5+a5*x5+b5*y5\neq 0\vee c5+\frac{1}{2}*a5*r+\frac{1}{2}*a5*x1+\frac{1}{2}*b5*y1\neq 0\vee\\
&c6+a6*x7+b6*y7\neq 0\vee c6+a6*x8+b6*y8\neq 0\vee c6+a6*\mu3+b6*\mu4\neq 0\vee\\
&c6+\frac{1}{2}*a6*x5+\frac{1}{2}*b6*y5\neq 0\vee c6+\frac{1}{2}*a6*r+\frac{1}{2}*a6*x2+\frac{1}{2}*b6*y2\neq 0\vee\\
&c7+\frac{1}{2}*a7*x10+\frac{1}{2}*b7*y10\neq 0\vee c7+\frac{1}{2}*a7*\mu3+\frac{1}{2}*b7*\mu4\neq 0\vee\\
&c7+\frac{1}{2}*a7*x5+\frac{1}{2}*a7*x6+\frac{1}{2}*b7*y5+\frac{1}{2}*b7*y6\neq 0\vee\\
&c7+\frac{1}{2}*a7*x5+\frac{1}{2}*b7*y5+\frac{1}{2}*a7*\mu1+\frac{1}{2}*b7*\mu2\neq 0\vee\\
&c7+\frac{1}{2}*a7*x8+\frac{1}{2}*a7*x9+\frac{1}{2}*b7*y8+\frac{1}{2}*b7*y9\neq 0\vee\\
&c8+a8*x3+b8*y3\neq 0\vee c8+a8*\mu1+b8*\mu2\neq 0\vee a5*r\neq 0)
\end{aligned}$$

(Formula (TrisectingAngle):(1).2): . . .

We now prove the above individual problems separately:

Proof of (Formula (TrisectingAngle): (1).1): . . .

This proof problem has the following structure:

(Formula(TrisectingAngle): (1).1.structure). . .

(Formula (TrisectingAngle): (1).1.structure) is equivalent to

(Formula (TrisectingAngle): (1).1.implication):. . .

(Formula (TrisectingAngle): (1).1.implication) is equivalent to

(Formula (TrisectingAngle): (1).1.not-exists): . . .

By introducing the slack variable(s)
$\{\xi\}$
(Formula (TrisectingAngle): (1).1.not-exists) is transformed into the equivalent formula

(Formula (TrisectingAngle): (1).1.not-exists-slack):...

Hence, we see that the proof problem is transformed into the question on whether or not a system of polynomial equations has a solution or not. This question can be answered by checking whether or not the (reduced) Groebner basis of
...
is exactly $\{1\}$.
Hence, we compute the Groebner basis for the following polynomial list:

$$\{-1+(-2)x10x9y10\ \xi + x10^2y9\ \xi + (-1)y10^2y9\ \xi, c8, x5, x8, -1+r\eta1, -1+a5^2\eta2+b5^2\eta2, -1+a6^2\eta3+b6^2\eta3, -1+a7^2\eta4+b7^2\eta4, -1+a8^2\eta5+b8^2\eta5, c5+\frac{b5\ r}{2}, -r+x4, -r+x7, b5\ r+(-1)b5\ x1+a5\ y1, b6\ r+(-1)b6\ x2+a6\ y2, b7\ x5+(-1)b7\ x6+(-1)a7\ y5+a7\ y6, b7\ x5+(-1)a7\ y5+(-1)b7\mu1+a7\mu2, b7\ x8+(-1)b7\ x9+(-1)a7\ y8+a7\ y9, -b6\ x5+a6\ y5, -b7\ x10+a7\ y10, -b7\ \mu3+a7\ \mu4, c5+a5\ x4+b5\ y4, c5+a5\ x5+b5\ y5, c5+\frac{a5\ r}{2}+\frac{a5\ x1}{2}+\frac{b5\ y1}{2}, c6+a6\ x7+b6\ y7, c6+a6\ x8+b6\ y8, c6+a6\ \mu3+b6\ \mu4, c6+\frac{a6\ x5}{2}+\frac{b6\ y5}{2}, c6+\frac{a6\ r}{2}+\frac{a6\ x2}{2}+\frac{b6\ y2}{2}, c7+\frac{a7\ x10}{2}+\frac{b7\ y10}{2}, c7+\frac{a7\ \mu3}{2}+\frac{b7\ \mu4}{2}, c7+\frac{a7\ x5}{2}+\frac{a7\ x6}{2}+\frac{b7\ y5}{2}+\frac{b7\ y6}{2}, c7+\frac{a7\ x5}{2}+\frac{b7\ y5}{2}+\frac{a7\ \mu1}{2}+\frac{b7\ \mu2}{2}, c7+\frac{a7\ x8}{2}+\frac{a7\ x9}{2}+\frac{b7\ y8}{2}+\frac{b7\ y9}{2}, c8+a8\ x3+b8\ y3, c8+a8\ \mu1+b8\ \mu2, a5\ r\}$$

The Groebner basis:
$\{1\}$
Hence, (Formula(TrisectingAngle):(1).1) is proved.

Proof of (Formula (TrisectingAngle): (1).2): ...
This proof problem has the following structure:

(Formula (TrisectingAngle): (1).2.structure): ...

(Formula (TrisectingAngle): (1).2.structure) is equivalent to

(Formula (TrisectingAngle): (1).2.implication) : ...

(Formula(TrisectingAngle): (1).2.implication) is equivalent to

(Formula (TrisectingAngle): (1).2.not-exists): ...

By introducing the slack variable(s)
$\{\xi 1\}$

(Formula (TrisectingAngle): (1).2.not-exists) is transformed into the equivalent formula

(Formula (TrisectingAngle): (1).2.not-exists-slack): ...

Hence, we see that the proof problem is transformed into the question on whether or not a system of polynomial equations has a solution or not. This question can be answered by checking whether or not the (reduced) Groebner basis of
...
is exactly $\{1\}$.
Hence, we compute the Groebner basis for the following polynomial list:

$$\{-1+(-3)x10^2x3y10\xi 1+x3y10^3\xi 1+x10^3y3\xi 1+(-3)x10y10^2y3\xi 1, c8, x5, x8, -1+r\ \eta 1, -1+a5^2\ \eta 2+b5^2\ \eta 2, -1+a6^2\ \eta 3+b6^2\ \eta 3, -1+a7^2\ \eta 4+b7^2\ \eta 4, -1+a8^2\ \eta 5+b8^2\ \eta 5, c5+\frac{b5r}{2}, -r+x4, -r+x7, b5\ r+(-1)b5\ x1+a5\ y1, b6\ r+(-1)b6\ x2+a6\ y2, b7\ x5+(-1)b7\ x6+(-1)a7\ y5+a7\ y6, b7\ x5+(-1)a7\ y5+(-1)b7\ \mu 1+a7\ \mu 2, b7\ x8+(-1)b7\ x9+(-1)a7\ y8+a7\ y9, -b6\ x5+a6\ y5, -b7\ x10+a7\ y10, -b7\ \mu 3+a7\ \mu 4, c5+a5\ x4+b5\ y4, c5+a5\ x5+b5\ y5, c5+\frac{a5\ r}{2}+\frac{a5\ x1}{2}+\frac{b5\ y1}{2}, c6+a6\ x7+b6\ y7, c6+a6\ x8+b6\ y8, c6+a6\ \mu 3+b6\ \mu 4, c6+\frac{a6\ x5}{2}+\frac{b6\ y5}{2}, c6+\frac{a6\ r}{2}+\frac{a6\ x2}{2}+\frac{b6\ y2}{2}, c7+\frac{a7\ x10}{2}+\frac{b7\ y10}{2}, c7+\frac{a7\ \mu 3}{2}+\frac{b7\ \mu 4}{2}, c7+\frac{a7\ x5}{2}+\frac{a7\ x6}{2}+\frac{b7\ y5}{2}+\frac{b7\ y6}{2}, c7+\frac{a7\ x5}{2}+\frac{b7\ y5}{2}+\frac{a7\ \mu 1}{2}+\frac{b7\ \mu 2}{2}, c7+\frac{a7\ x8}{2}+\frac{a7\ x9}{2}+\frac{b7\ y8}{2}+\frac{b7\ y9}{2}, c8+a8\ x3+b8\ y3, c8+a8\ \mu 1+b8\ \mu 2, a5\ r\}$$

The Groebner basis:
$\{1\}$
Hence, (Formula(TrisectingAngle):(1).2) is proved.
Since all of the individual subtheorems are proved, the original formula is proved.

□

5. Conclusion We have shown the computer origami construction with the example of trisecting an angle. Our computational origami system not only does the simulation of origami folds [Ida 2003], but also proves the correctness of the construction by accessing the implementation of the Gröbner bases algorithm [Buchberger 1970] and the implementation of the above decision algorithm in Theorema [Buchberger 2000]. The premise polynomial set is not optimal since it contains extra geometrical constraints unnecessary for proving the theorem. We are currently working to minimize the premise polynomial set.

As a next step of our research we plan to study origami solving problem, which asks for finding a sequence of origami steps that will lead to an origami object with a desired property. However, it is clear that this problem is analogous to the problem of finding geometric objects with desired properties using only a ruler and a compass. Note, however, that the two problems - origami construction and the ruler-and-compass construction - are not equivalent. As we have seen, the trisection of an angle is possible by origami but not by the ruler-and-compass method. For further development of origami construction, in analogy to the ruler-and-compass construction problem, Galois theory suggests itself as the main approach to solving the origami construction problem.

REFERENCES

B. Buchberger, Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems (An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations). Aequationes mathematicae 4/3, 1970, pp. 374-383. (English translation in: Buchberger, B., and Winkler, F. (eds.), Gröbner Bases and Applications, Proceedings of the International Conference "33 Years of Groebner Bases", 1998, RISC, Austria, London Mathematical Society Lecture Note Series, Vol. 251, Cambridge University Press, 1998, pp. 535 -545)

B. Buchberger, C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, K., D. Vasaru, W. Windsteiger, The Theorema Project: A Progress Report, In: Symbolic Computation and Automated Reasoning (Proceedings of CALCULEMUS 2000, Symposium on the Integration of Symbolic Computation and Mechanized Reasoning, August 6-7, 2000, St. Andrews, Scotland), Kerber, M. and Kohlhase, M. (eds.), A.K. Peters, Natick, Massachusetts, pp. 98-113.

B. Buchberger, T. Ida, Origami Theorem Proving, SFB Scientific Computing Technical Report 2003-23-Oct, Johannes Kepler University RISC, 2003.

T.L. Chen, Proof of the impossibility of trisecting an angle with Euclidean tools, Mathematics Magazine Vol. 39, pp. 239-241, 1966.

K. Fushimi, Science of Origami, a supplement to Saiensu, Oct. 1980, p. 8 (in Japanese).

R. Geretschläger, Geometric Constructions in Origami (in Japanese, translation by Hidetoshi Fukagawa), Morikita Publishing Co., 2002.

K. Haga, Origamics Part I: Fold a Square Piece of Paper and Make Geometrical Figures (in Japanese), Nihon Hyoronsha, 1999.

T. Hull, Origami and Geometric Constructions, available online at http://web.merrimack.edu/ thull/geoconst.html, 1997.

H. Huzita, Axiomatic Development of Origami Geometry, Proceedings of the First International Meeting of Origami Science and Technology, pp. 143-158, 1989.

T. Ida, M. Marin, and H. Takahashi, Constraint Functional Logic Programming for Origami Construction, Proceedings of the First Asian Symposium on Programming Languages and Systems (APLAS2003), Lecture Notes in Computer Science, Vol. 2895, pp 73-88, 2003.

S. Wolfram, The Mathematica Book, 5th ed., Wolfram Media, 2003.

Department of Computer Science,
University of Tsukuba
305-8573,Japan
E-mail: ida@cs.tsukuba.ac.jp

Research Institute for Symbolic Computation,
Johannes Kepler University,
A4040 Linz, Austria
E-mail: buchberger@risc.uni-linz.ac.at