
5.0 The Tree Method of Origami Design

This section describes the mathematical ideas that underlie (訳注：underlie?) the tree method of origami design —and technical origami, or *origami sekkei*— in particular. The Japanese art of origami, though centuries old, has undergone a renaissance in the last few decades from new advances in origami design. The simple stylized birds, flowers, and fish of old have been replaced by incredibly complex replicas of insects, crustacea, mammals, amphibians, dinosaurs, humans, objects, and just about everything under the sun. The explosion of subject matter was driven in part by new discoveries in design techniques. One of the most significant of these discoveries was a series of mathematical algorithms for designing origami bases in terms of the number, size, and arrangement of their points. This document section some of those algorithms.

このセクションは特に、Tree Method と呼ばれる折り紙の創作手法（そして折り紙設計）の基礎となる数学的な考え方について説明する。日本の芸術である折り紙には数世紀の歴史が存在するが、ここ数十年の間に、折り紙の創作における新手法によってルネッサンスが起きている。単純に様式化された古くからの鳥、花、および魚が、昆虫、甲殻類、哺乳動物、両生類、恐竜、人間、物、および世界中のあらゆるものの信じられないほど複雑なレプリカに取って代わられた。この爆発的な主題の増加は、一部には新たな創作手法の発見によってもたらされた。これらの発見で最も重要なものの1つは、基本形のカドの数、大きさ、並べ方を定めるという意味での基本形創作のための、一連の数学的なアルゴリズムであった。このセクションではそれらのアルゴリズムのいくつかについて述べる。

Even before mathematics came into the picture, the concept of “points” — flaps that become appendages of an origami model — and their number, length, and connections have figured heavily in the origami designer's approach to creating new folds. Over the past few years, I and several folders have been formulating mathematical techniques for designing origami models on the basis of their points. We have discovered several general algorithms for design, algorithms that enable one to compute the crease pattern for a base containing any number of points of arbitrary complexity. A complete description of such algorithms and all of their nuances would fill a book — in fact, I am working on that very task — but the basic technique can be quickly outlined and may be of interest to folders interested in exploring new bases and symmetries.

数学的手法が登場する以前から、「カド」の概念、つまり折り紙作品において突起となる部分と、その数、長さ、つながりなどが、新しい折り紙を創作しようとする折り紙作家にとって重要な役割を演じてきた。ここ何年かにわたり、私を含め何人かの作家が、折り紙作品をカドに基づいて創作するための数学的技法を定式化してきた。その結果、創作に関していくつかの一般的なアルゴリズムが明らかとなった。それは任意の複雑性を持つ任意の数のカドを備えた基本形の展開図を計算することを可能とするアルゴリズムである。そういったアルゴリズムを一つ一つ詳細に説明しようと思ったら本が一冊書けてしまうが（実は私はその執筆をしているところである（訳注：「Origami Design Secrets: Mathematical Methods for an Ancient Art」として A K Peters Ltd. から 2003 年 10 月に出版））、基本的な手法を説明するのにはそれほど時間はかからない。これは常日頃から新しい基本形と均整美を探究している折り紙作家にとって、興味深い内容となるだろう。

I am going to describe an algorithm using terminology and concepts that I developed in my own research. However, I should acknowledge major contributions from other mathematical folders. In any scientific endeavor, the rate of progress is vastly accelerated through the interaction and cross-fertilization of ideas among several investigators and those who have contributed to this theory are many: they include Jun Maekawa, Fumiaki Kawahata, Toshiyuki Meguro, Alex Bateman and Tom Hull, with additional insights from John Montroll, Peter Engel, Issei Yoshino, Brian Ewins, Jeremy Shafer, and Martin and Erik Demaine. My thanks and acknowledgment to these, as well as the many more folders whose works formed both inspiration and guide for my work over the years.

ここでは、自身の研究の中で発展させた用語と概念を用いた新手法について述べるつもりだ。しかしながら、他の数理的折り紙作家による大きな貢献については謝辞を述べなければならない。どんな科学的な試みも、さまざまな研究者のあいだで相互に影響し合い知識を交換し合うことで、その進捗速度を限りなく加速させる

ことができる。そして、この新手法において貢献を果たした人も数多い。その中には、前川淳、川畑文昭、目黒俊幸、Alex Bateman、Tom Hull が含まれ、John Montroll、Peter Engel、吉野一生、Brian Ewins、Jeremy Shafer、Martin Demaine、Erik Demaine らの洞察からも得るところがあった。これらの人たちに、そして何年にもわたって自分の作品にとっての示唆と指針となる作品を生み出した他の多くの折り紙作家たちに、感謝と謝辞を述べる。

And now, on with design.

それではこれから、折り紙の創作について述べる。

The name “technical folding” has come to be applied to origami designs that, whether intentionally or not, display a strong geometrical elegance and sophistication. The target concept for much of technical folding is the **base**. A base is simply a geometric shape that resembles the subject to be folded — usually, a shape having the same number and length of flaps as the subject has appendages. For example, a base for a bird might have four flaps, corresponding to ahead, tail, and two wings. A slightly more complicated subject such as a lizard would require a base with six flaps, corresponding to the head, four legs, and a tail. And an extremely complicated subject such as a flying horned beetle might have six legs, four wings, three horns, two antennae and an abdomen, requiring a base with sixteen flaps. You needn’t stop there, either. Throw in mouthparts and a separate flap for the thorax, if you like, for a total of nineteen flaps. The number of flaps required in a model depends on the level of anatomical accuracy desired by the designer. Historically, much origami design was performed by trial and error — manipulating a piece of paper until it began to resemble something recognizable. However, for a complex subject, trial and error is a highly inefficient approach, since one is unlikely to stumble upon a nineteen-pointed base with flaps of the right size in the right places purely by luck. For such models, a more directed approach is called for.

意図的か否かにかかわらず幾何学的な優雅さと洗練さを強く示すような折り紙作品に対して、「折り紙設計」という言葉が使われるようになった。折り紙設計が主な対象としているのは**基本形**である。基本形とは、簡単に言えば折る対象に似た幾何学形状であり、一般的には、対象が有する突起部分と同じ数と長さのカドを有する形である。例えば鳥の基本形には、頭、尾、二つの翼に対応する四つのカドがあるだろう。トカゲのようなもう少し複雑な対象の場合、頭、四本の脚、尾に対応する六つのカドが必要となるかもしれない。そして空を飛んでいるカブトムシのような非常に複雑な形をした対象の場合、六本の脚、四枚の羽根、三本の角、二本の触覚、そして腹部の 16 のカドが基本形に必要となるかもしれない。それにとどめる必要も無い。もし望むのなら、顎に二つ、胸部のために独立したカドを一つ追加すれば、合計 19 のカドになる。作品の中で必要となるカドの数は、作家の望む形状の正確さの水準に依存する。これまでは、多くの折り紙の創作は試行錯誤によってなされてきた。つまり、認識可能な何かに似始めるまで、一枚の紙に対して手を加え続けるということである。しかし、複雑な対象に対しては試行錯誤は非常に効率の悪いアプローチである。なぜなら、正しい位置に正しい大きさで 19 個のカドのある基本形を、純粋な幸運によって偶然発見するということは、ありそうもない話であるからだ。そのような作品に対してはより直接的なアプローチが求められる。

For a given number and distribution of flaps, there are always many possible bases that can be folded that differ in various ways: the overall size of the base, which flaps have how many layers, how the flaps are positioned in the plane of the paper, the stacking order of the layers, and so forth. I doubt that there is a universal algorithm that could come up with every possible base for a given subject; at any rate, most folders would be satisfied with an algorithm that could at least come up with one. It turns out that such an algorithm exists for a particular class of base. We will consider bases that have a special property: their flaps and layers can be arranged such that (1) one edge of every flap lies in a common plane and (2) when the edges all lie in a plane, all of the layers of the base are perpendicular to the plane. I call this type of base a **uniaxial base** (because when the base is flattened, all of the flaps lie along a single common axis). An example of a hypothetical uniaxial base for a hypothetical four-legged (and hence six-flapped) creature is shown in figure 5.1.

カドの数と配置が与えられたとき、折ることが可能な展開図は常に多数存在し、それらは以下のような様々な点で異なる。すなわち、基本形の全体的な大きさ、どのカドが何枚の層になっているか、紙の中でカドがどのように配置されているか、面の重ね順などである。与えられた対象形状について、可能な基本形をすべて見つけることが出来る万能なアルゴリズムが存在するかどうかということについては、私は否定的だ。いずれにせよ、ほとんどの作家は、少なくとも一つの基本形が求められるアルゴリズムがあれば満足するだろう。実際、ある特定の種類の展開図に対してはそのようなアルゴリズムが存在する。ここでは、次のような条件を満たす

ようにカドと面を配置できるという特別な性質を持つ基本形について考える。(1) 全てのカドの稜線の一つを共通の平面に載せることができ、(2) そのように稜線の全てを同一平面にのせたとき、基本形の全ての面がその平面に垂直となる。ここではこのタイプの基本形を**単軸基本形**と呼ぶ(なぜなら、その基本形を平らにつぶしたとき、全てのカドが単一の共通軸上に並ぶためである)。仮に四本の脚をもつ(したがって、六個のカドがある)生物の場合、その仮想的な単軸基本形は、例えば図 5.1 の様になる。

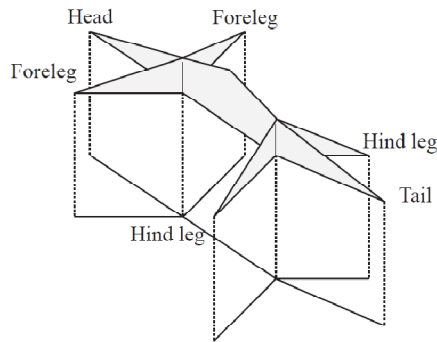


Figure 5.1. Schematic of a uniaxial base for an animal with four legs, a head, body, and tail. It's a uniaxial base if it can be manipulated so that all of the flaps lie in a plane and all of the layers are perpendicular to the plane. The shadow of the base consists entirely of lines.

図 5.1 四本の脚と頭、体、尾が一つずつある動物の単軸基本形の概略図。全てのカドが同一平面上に載り、かつ全ての面がその平面に対して垂直になるように基本形を変形できる場合、それを単軸基本形と呼ぶ。基本形の影は複数の直線のみからなる。

While a great many origami bases can be represented as uniaxial bases including the four classic bases — the kite, fish, bird, and frog bases — many others cannot. For example, John Montroll's Dog Base and its derivatives do not fall into this category. But there are many bases that do, and most importantly from the standpoint of design, for a great many subjects, a uniaxial base is a sufficiently good approximation of the subject that you can easily transform it into a very accurate and detailed replica of the subject.

四つの古典的な基本形、すなわち凧形基本形、魚の基本形、鶴の基本形、蛙の基本形を含む非常に多くの基本形は単軸基本形だと言える一方、単軸基本形とは言えない基本形も多い。例えば John Montroll による犬の基本形およびその派生型はこのカテゴリーに含まれない。しかし、このカテゴリーに含まれる基本形も多数存在しているし、創作の見地からみて最も重要なことに、非常に多くの対象物について、単軸基本形は十分ふさわしい近似形状であり、対象物の正確で詳細なレプリカへ簡単に変形できる。

We can represent any uniaxial base by the stick figure formed by the projection of the base into the plane, i.e., the shadow cast by the base if illuminated from above. This stick figure has the same number and length sticks as the base has flaps, and therefore it is a convenient abstraction of the base, even as the base is an abstraction of the subject which we are trying to fold. The problem to be solved is: for a given stick figure, can one find a way of folding a square into a base whose projection is that stick figure?

どのような単軸基本形でも、基本形を平面に投影して形作られる木構造(すなわち、基本形に上から光を当てた際に生じる影)によって表現することができる。この木構造には、基本形のカドと同じ数と長さの枝がある。そのためこの木構造は、折ろうとしている対象物を抽象化した基本形を、さらに扱いやすく抽象化したものと言える。そこで、解決すべき課題は、ある木構造が与えられたとき、その木構造へ投影されるような基本形に正方形を折り畳むための方法を探すことである。

Let's first define some terms describing this stick figure. The mathematical term for a stick figure is a **graph**. I'll call the particular stick figure for a uniaxial base the graph of the uniaxial base. Each line segment in the graph is called an **edge** of the graph; each endpoint of a line segment (or point where two segments come together) is called a **node**. Each flap of the base corresponds to a distinct edge of the graph, and since each flap has a specified length, each edge of the graph has a length as well, which is the length

of its associated flap. Thus, the planar graph for the six-legged animal base shown in figure 5.2 has eight nodes and seven edges. In figure 2, I have assigned a length of 1 to each edge of the graph, which implies that each segment in the base has the same length. One can, of course, assign any desired combination of lengths — if you wanted an extra-long tail, you could give it a length of 2 or 3 — but for simplicity, we'll make all of the edges the same in this example.

まず最初に、この木構造を説明するためにいくつかの用語を定義する。このような木構造は数学用語で**グラフ**という。ここでは特に単軸基本形の木構造のことを、単軸基本形のグラフと呼ぶ。グラフ中のそれぞれの線分をグラフの**辺**と呼び、線分の端点（または二本の線分が交差する場所）を**節点**と呼ぶ。基本形の各カドはグラフ中のそれぞれの辺に対応する。基本形のカドにはそれぞれ決まった長さがあるので、グラフの辺にも長さがあり、それは対応するカドの長さと同じである。したがって、6つのカドを持つ動物の基本形の場合、二次元グラフは図5.2に示すように8つの節点と7つの辺を持つ。この図において、私はグラフの各辺に1の長さを割り当てた。それはすなわち、対応する基本形のすべてのカドが同じ長さを持つということである。もちろん、任意の長さの組み合わせを割り当てることは可能である。もしとても長い尾が必要なら、2や3といった長さを割り当てることが出来る。しかしこの例では、話を簡単にするため、すべての辺の長さは等しいとする。

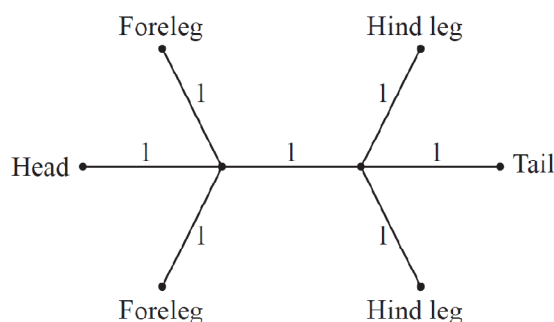


Figure 5.2. Planar graph for the six-legged base. Each edge of the graph has a length of 1 unit in this example.

図 5.2 六つのカドを持つ基本形の二次元グラフ。この例ではグラフの各辺の長さは1単位長である。

We will also restrict ourselves to bases that don't contain any loops of paper, which correspond to planar graphs that don't have any cycles in them. If a base has a closed loop, that implies it has a hole in it somewhere, which would imply cutting — a no-no, in origami design. (Of course, we can always simulate a loop by joining two flaps together and hiding the joint.) A graph with no cycles is called a **tree graph** or just **tree**. It's easy to see that for any tree, the number of nodes is always exactly one more than the number of edges. Thus, we will be searching for bases whose projections are trees; for this reason, I call this design approach the *tree method* of design.

また、基本形は紙のループ構造を含まないものに限定する。それに対応して、二次元グラフは環をもたないものに限定する。もし基本形が閉じたループを含んでいた場合はどこかに穴があることを意味するが、それはすなわち紙を切るということであり、折り紙の創作においては禁じ手である（もちろん、二つのカドを接続させて接合部を隠すことにより、ループ構造を装うことはいつでも可能である）。環のないグラフは、**木グラフ**または単に**木**と呼ばれる。どのような木についても、節点の数が辺の数よりちょうど一つだけ多いことは容易に見て取れる。そして、ここでは写像が木となる基本形を探すこととする。そのためこの創作手法をツリーメソッドと呼ぶ。

We will also divide the nodes into two types: **terminal nodes** are nodes that come at the end of a single edge. Terminal nodes correspond to the tips of legs, wings, and other appendages. Nodes formed where two or more edges come together are called **internal nodes**.

また、ここでは節点を次の二つの種類に分類する。**端節点**はある一辺の終端に位置する節点である。端節点は脚、翼、およびその他のとがった場所の先端に対応する。二つ以上の辺が交差する節点を**内節点**と呼ぶ。

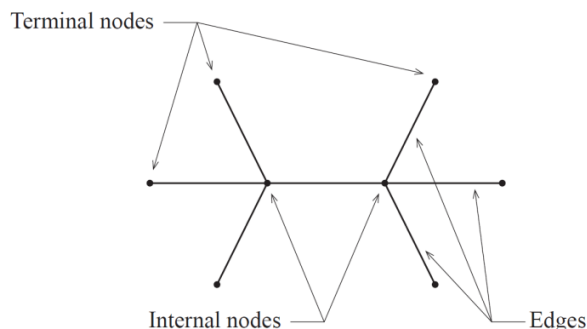


Figure 5.3. Parts of a tree graph. Line segments are called edges of the graph; points where lines end or come together are nodes. Terminal nodes are nodes with only one edge; internal nodes have two or more edges.

図 5.3 木グラフの要素。線分はグラフの辺と呼ばれ、線分の終端または線分と線分との交点は節点である。端節点は 1 つの辺だけが接続された節点で、内節点には 2 つ以上の辺が接続される。

Suppose we have a base that has been folded from a square and we construct its tree. If we unfold the base, we get a square with a crease pattern that uniquely defines the base. The folding sequence that transforms the square into the base can be thought of as a mapping between points on the square and points on the tree.

正方形から折った基本形があるとして、その基本形の木を構築することを考える。基本形を広げると、展開図のついた正方形が得られるが、その展開図が基本形を一意に定義する。正方形を基本形へと変形するための折り手順は、正方形上の点と木上の点との間の写像と考えることができる。

If you think of the tree as the shadow of the base, you can see from figure 1 that wherever you have vertical layers of paper, there are many points on the base that map to the same point on the tree. However, at the terminal nodes of the tree, there is exactly one point on the square that maps to the node. Thus, we can uniquely identify the points on the square that become the tips of the flaps of the base.

木を基本形の射影であると考えた場合、図 1 から、垂直な紙の面があるところではどこでも、基本形上の多数の点が木上の同一の点に対応することが見て取れる。しかしながら、木の端節点については対応する正方形上の点はただ一つしか存在しない。したがって、基本形のカドの先端になる正方形上の点を一意に特定することが可能である。

Now consider the following thought experiment: suppose an ant were sitting at the tip of one of the legs of the base and wished to travel to another part of the base — say, the tail — without leaving the paper. It would have to walk down the leg to the body, down the body, and back out the tail. The distance it traveled would be (length of the leg) + (length of the body) + (length of the tail).

ここで、次のような思考実験を行う。基本形の脚のひとつの先端にアリがいて、その紙から離れないで、基本形の別の場所（例えば尾）へと移動しようとしているとする。その場合、脚を胴体へ向かって歩き、胴体を通り過ぎて、尾へと抜けるようにしなければならないだろう。その行程の長さは（脚の長さ）+（胴体の長さ）+（尾の長さ）になるだろう。

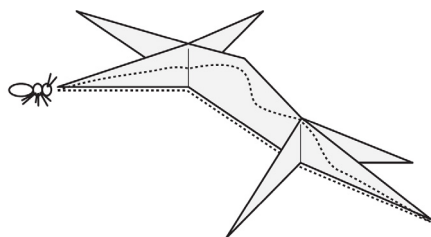


Figure 5.4. An ant wishes to go from a foreleg to the tail along the base. It can take several different paths, but the most direct path is the path that lies in the plane of projection.

図 5.4 アリは基本形に沿って前脚から尾にむかって行こうとしている。いくつかの異なる経路をとることが可能であるが、最短の経路は射影平面上の経路である。

Now, let's think about what the path of the ant would look like on the unfolded square (you can imagine dipping the ant into ink so that it left a trail soaking through the paper as it walked). On the square, the path might meander around a bit or it might go directly from one point to the other, depending on the course of the ant. One thing is certain, however: the distance traveled on the unfolded square must be at least as long as the minimum distance traveled along the base.

ここで、アリの経路が展開した正方形においてどのように見えるか考えてみよう（アリをインクに浸して、その足跡が紙の上に残っているという場面を想像すればよい）。正方形上の経路はアリのとった行程に依存し、少し曲がっているかもしれないし、ある点からある点まで一直線になっているかもしれない。しかし、次の点は明らかである。展開した正方形上での行程の長さは、基本形に沿った最短の行程の長さと同じか、それよりも長くなければならない。

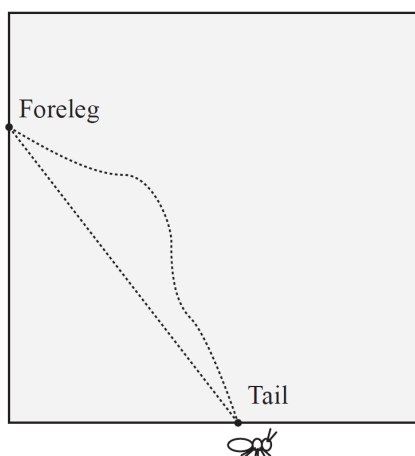


Figure 5.5. The trail of the ant. The path of the ant might wander around on the square once you've unfolded it, but there is no way that the path can be shorter on the square than the path was on the base. Thus, the distance between two nodes on the square must be at least as large as the distance between the two nodes measured along the edges of the tree.

図 5.5 アリの行程。広げられた正方形上でのアリの経路が曲がりくねることはあり得るが、基本形での経路よりも正方形上での経路が短くなることはあり得ない。したがって、正方形上での二つの節点間の距離は、木の辺に沿ってその二つの節点間の距離を測ったものと同じ長さか、それよりも長くなければならない。

This illustrates an extremely important property of any mapping from a square to a base: the distance between *any* two points on the square must be greater than or equal to the distance between the two corresponding points on the base. And in particular, this relationship must hold for *any* two points on the base that correspond to nodes on the tree. Now while this condition must hold for any pair of points on the base, it turns out that if it holds for any pair of terminal nodes, it will hold for *every* pair of points on the base. That is, if you identify a set of points on the square corresponding to terminal nodes of a tree and the points satisfy the condition that the distance between any pair of points on the square is greater than or equal to the distance between the points as measured on the graph, then *it is guaranteed that a crease pattern exists to transform the square into a base whose projection is the tree*.

これは、正方形と基本形との任意の関連付けが満たすべき重要な特性を示している。すなわち、正方形上の任意の二点間の距離は、基本形上での対応する二点間の距離以上でなければならない。特に、木の節点に対応する展開図上の任意の二点について、この関係が成立しなければならない。ここで、この条件が基本形上の任意

の点の組で成立しなければならないと言っても、もしこの条件が任意の端節点の組について成立していれば、基本形上のあらゆる点の組においても成立するということが分かる。すなわち、ある木の端節点に対応する正方形上の点の集合を特定したとき、それらの点について、正方形上における任意の点の組の間の距離がグラフ上の点の間の距離以上であるという条件が満たされているならば、その木に対応する基本形へ正方形を変形するような展開図が存在するということが保証される。

This is a remarkable property. It tells us that no matter how complex a base is sought, no matter how many points it may have and how they are connected to one another, we can *always* find a crease pattern that transforms the square (or any other shape paper, for that matter) into the base. Putting this into mathematical language, we arrive at the fundamental theorem of the tree method of design (which I call the “tree theorem” for short):

これは特筆すべき特性である。ここから分かることは、必要とされる基本形がどれだけ複雑であっても、それがどれだけ多くのカドを持っていたとしても、そしてそれらのカドが互いにどのように接続していたとしても、正方形（ついでに言えば他の任意の形状）をその基本形へと変形する展開図を常に見つけることが出来るということである。これを数学的な言葉で表すと、ツリーメソッドによる創作手法の基本的な定理（略して「ツリー定理」と呼ぶ）が得られる。

Define a simply connected tree P with terminal nodes $P_i, i=1, 2, \dots, N$. Define by l_{ij} the distance between nodes P_i and P_j as measured along the edges of the tree; that is, l_{ij} is the sum of the lengths of all the edges between nodes P_i and P_j . Define a set of points \mathbf{u}_i in the unit square $u_{i,x} \in [0,1], u_{i,y} \in [0,1]$. Then a crease pattern exists that transforms the unit square into a uniaxial base whose projection is P if and only if $|\mathbf{u}_i - \mathbf{u}_j| \geq l_{ij}$ for every i, j . Furthermore, in such a base, P_i is the projection of \mathbf{u}_i for all i .

端節点 $P_i (i=1, 2, \dots, N)$ を持つ単純グラフである木 P を定義する。そして P_i, P_j の間の距離を木の辺にそって計測した長さを l_{ij} と定義する。すなわち、 l_{ij} は節点 P_i と P_j の間の全ての辺の長さの合計である。単位正方形の中に点の集合 \mathbf{u}_i を定義する ($u_{i,x} \in [0,1], u_{i,y} \in [0,1]$)。このとき、写像が木 P となるような単軸基本形へ単位正方形を変形する展開図が存在するのは、すべての i, j において $|\mathbf{u}_i - \mathbf{u}_j| \geq l_{ij}$ (訳注: $|\mathbf{u}_i - \mathbf{u}_j| \geq l_{ij}$ の間違い?) であるときであり、そのときに限る。さらに、そのような基本形においては、全ての i について P_i は \mathbf{u}_i の写像である。

Although the proof of the tree theorem is beyond the scope of this article, we won't let that stop us from using it. The tree theorem tells us that if we can find a distribution of nodes on a square for which the distance between any two nodes is greater than or equal to the distance between the nodes on the tree, then a crease pattern exists that can transform that node pattern into a base. For a given tree, there are often several possible node patterns that satisfy the tree theorem, each of which yields a different base. For our six-pointed base, a little doodling with pen and paper will reveal that the pattern of nodes shown in figure 5.6 fits inside of a square of side $2\sqrt{((121+8\sqrt{179})/65)} \approx 3.7460$.

ツリー定理の証明はこの文章の範疇を超えるが[1]、だからといってそれを使っていけないことはない。ツリー定理によれば、正方形上の節点の配置について、任意の二つの節点間の距離が木上で対応する節点間の距離以上である場合、そのような節点のパターンを基本形へと変形可能な展開図が存在する。木が与えられたとき、ツリー定理を満たすような可能な節点パターンはしばしば複数存在し、それらは異なった基本形をもたらす。上で挙げた6つのカドを持つ基本形について、鉛筆と紙で少し計算してみると、図5.6に示す節点パターンが、一辺の長さ $2\sqrt{((121+8\sqrt{179})/65)} \approx 3.7460$ の正方形の内側に収まることが分かる。

- [1] For a proof of the tree theorem, see “A Computational Algorithms for Origami Design,” 12th Annual Symposium on Computational Geometry, May 22-26, 1996, Philadelphia, PA.
ツリー定理の証明に関しては、“A Computational Algorithms for Origami Design,” 12th Annual Symposium on Computational Geometry, May 22-26, 1996, Philadelphia, PA を参照のこと。

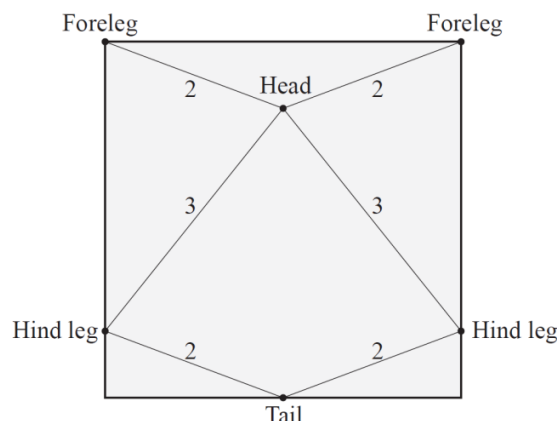


Figure 5.6. Node pattern that satisfies the tree theorem for the six-legged tree.

図 5.6 6つのカドを持つ木における、ツリー定理を満たすような節点パターン

The tree theorem is an existence theorem; it says that a crease pattern exists but it doesn't tell us what this supposed crease pattern actually is. However, it does provide a strong clue. The tree theorem says that the points on the square that correspond to terminal nodes of the tree become the tips of the flaps on the folded base. Are there any other parts of the square that we can identify on the base?

ツリー定理は存在定理である。ある展開図が存在するということはあっても、その存在するとされる展開図が実際にはどのようなものであるかについては言及しない。しかし、この定理はとてもおおきな手がかりとなる。ツリー定理によれば、木の端節点に対応する正方形上の点は、折り畳まれた基本形のカドの頂点になる。では、正方形上の他の箇所、基本形上の位置を特定できる箇所はあるだろうか？

Consider the inequality in the tree theorem. Two points corresponding to terminal nodes must be separated on the square by a distance *greater than or equal to* the distance between them on the tree. In the special case where equality holds, we can uniquely identify the line between the two points. We will call a line on the square that runs between any two terminal nodes a **path**. Every path has a **minimum length**, which is the sum of the edges of the tree between the terminal nodes that define the path. (In the symbolism of the tree theorem, l_{ij} is the minimum length of path ij .) The **actual length** of a path is given by the distance between the terminal nodes as measured upon the square ($|\mathbf{u}_i - \mathbf{u}_j|$ in the tree theorem). Any path for which its actual length is equal to its minimum length is called an **active path**.

ツリー定理の式が不等式であることに注意しよう。二つの端節点に対応する正方形上の二つの点の間の距離は、木上での二点間の距離以上でなければならない。等号が成り立つような特別な場合では、二点間を結ぶ線を一意に決定することが出来る。ここで、正方形上の任意の二つの端節点の間を結ぶ線を**パス**と呼ぶことにする。それぞれのパスは**最小長**を持ち、その値はそのパスを定義する端節点間の木上の辺の長さの合計である。(ツリー定理の表記における l_{ij} が、パス ij の最小長である。) パスの**実長**は、正方形上で測定した端節点間の距離として与えられる(ツリー定理では $|\mathbf{u}_i - \mathbf{u}_j|$ である)。任意のパスについて、実長が最小長に等しいとき、そのパスを**実効パス**と呼ぶ。

On the base, the only route between two flap tips that is equal to the distance between the terminal nodes lies in the plane of the projection. Thus, any active path between two terminal nodes on the square becomes an edge of the base that lies in the plane of projection. Consequently, we have another important result:

基本形において、二つのカドの頂点間の経路であって、その距離が端節点間の距離と等しくなるようなただ一つの経路は、射影平面上に存在する。したがって、正方形上の二端節点の間の任意の実効パスは、基本形において射影平面に乗る稜線となる。その結果、別の重要な性質が明らかとなる：

All active paths between terminal nodes map to an edge of the base in the plane

of projection of the base.

端節点間の全ての実効パスは、基本形の射影平面上にある稜線にマッピングされる。

Active paths on the square lie in the plane of projection of the square, but the plane of projection is where the vertical layers of paper in the base meet each other. Thus, active paths are not only edges of the base: they are major creases of the base.

正方形上の実効パスはその正方形の射影平面上に乗るが、その射影平面において、基本形の垂直な面が互いに接合している。したがって、実効パスは基本形の稜線であるだけでなく、基本形の主要な折り線である。

So now we have the rudiments of the crease pattern for the base. We know the points on the square that correspond to terminal nodes of the tree become the tips of the flaps of the base, and we know that active paths on the square become major creases of the base. Furthermore, they are creases that lie in the plane of projection of the base. In many cases, this information alone is enough to find the complete crease pattern for the uniaxial base that corresponds to the tree. In many cases, though, it isn't. So we'll keep going.

そしてこれが基本形の展開図の基礎となる。すでに見たように、木の端節点に対応する正方形上の点は基本形のカードの頂点になり、正方形上の実効パスはその基本形の主要な折り線になる。その上、それらの折り線は基本形の射影平面上に乗る。多くの場合、この情報だけでも木に対応する単軸基本形の完全な展開図を見つけるのに十分であるが、そうでない場合も多い。そのためより深く考察してみよう。

If you draw all of the active paths on a square, you get a network of lines, each of which is a major crease of the base. Figure 6 includes the active paths for the node pattern for the six-legged animal, labeled with their length. Each of those lines will turn out to be a crease in the base.

正方形上にすべての実効パスを描くと、それらは線のネットワークとなり、それぞれの線が基本形の主要な折り線となる。図 5.6 には、6 つのカードを持つ動物の節点パターンに対する実効パスが含まれており、それぞれの長さが記されている。それぞれの線が基本形の折り線になる。

Since the active paths map one-to-one to paths on the tree, any features on the tree can be mapped directly onto the active paths. Specifically, we can locate where on an active path each internal node falls. If our hypothetical ant travels from one terminal node to another encountering internal nodes at distances d_1 , d_2 , d_3 , and so forth along the way, then when we draw the crease pattern, we can identify each internal node along the active path connecting the nodes at the same distances from the terminal nodes. Thus, we can add all of the internal nodes to our budding crease pattern. In figure 5.7, I've identified all of the nodes, terminal and internal, by a letter on the tree, and have added their locations to the active paths in the crease pattern on the square. Note that in general, an internal node will show up on more than one active path.

それぞれの実効パスは一つ一つ木上のパスと対応しているため、その木に関する任意の特徴を直接実効パスに関連付けることができる。特に、それぞれの内節点の実効パス上のどこに対応するのかを決定することができる。前述の仮想的なアリがある端節点から別の端節点へ移動するとき、経路に沿った距離が d_1 , d_2 , d_3 ... のところで内節点を通過したとすると、展開図を描いた時、それらの内節点の位置は、両端節点を結んだ実効パス上で、端節点からの距離が経路に沿った距離と等しい場所であると決定することができる。そのため、先ほどまで作りかけている展開図にすべての内節点を追加することが可能である。図 5.7 において、木上に文字を付加することですべての端・内節点を識別し、そしてそれらの場所を正方形上の展開図にある実効パスに追加した。なお、一般的に内節点は複数の実効パス上に現れることを注記しておく。

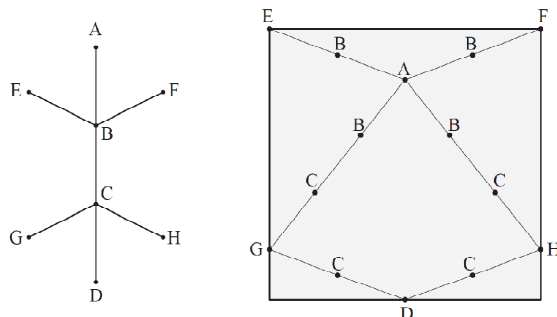


Figure 5.7. (Left). Tree with all nodes lettered. (Right) Crease pattern with terminal nodes, internal nodes, and active paths.

図 5.7. (左) 全ての節点に文字を付加した木。(右) 端節点、内節点、実効パスを備えた展開図。

It's also worth pointing out that we don't show any terminal nodes along the edges of the square because the paths between nodes G and E , E and F , and F and H are not active paths.

また、正方形の輪郭上にいかなる内節点も示さなかったことも特記しておく。なぜなら G - E 、 E - F 、 F - H の節点間のパスは実効パスでないためである。

It can be shown that active paths never cross each other, so the only places where lines come together on the square are terminal nodes. This pattern of lines breaks up the square into a set of polygons. In some of the polygons, all of their sides are active paths (like the inverted-kite-shaped quadrilateral in the center of figures 5.6 and 5.7). If one of the sides of a polygon lies on the edge of a square, it may or may not be an active path (in figure 5.6 and 5.7, each triangle has one side on the edge of the square that is not an active path.)

実効パスは決して互いに交差しないことを示すことができ、そのため正方形上で直線が交差する唯一の場所は端節点である。これらの直線によるパターンは正方形を多角形の集合に分割する。いくつかの多角形においては、すべての辺が実効パスとなる（図 5.6 や 5.7 の中心にある逆風形の四角領域など）。もし多角形の辺の一つが正方形の端にある場合、それは実効パスかもしれないしそうでないかもしれない（図 5.6 および 5.7 で、各三角形は正方形の端に重なる辺の一つを持っており、それは実効パスではない）。

We'll call a polygon whose corners are terminal nodes and whose sides are active paths or the edge of the square an **active polygon**. Each active polygon has the property that all of its sides map to the plane of projection of the base when the square is folded into a base. Consequently, to find a crease pattern that collapses the square into the base, it is necessary to find a crease pattern that maps the network of active paths onto the plane of projection of the base.

ここで、多角形の頂点が端節点であり辺が実効パスまたは正方形の輪郭である場合、そのような多角形を**実効多角形**と呼ぶことにする。それぞれの実効多角形は、正方形が基本形へ折りたたまれた時、基本形の射影平面にすべての辺をマッピングできるという特徴を持つ。よって正方形を基本形に畳めるような展開図を探すためには、実効パスの集合を基本形の射影平面にマッピングするような展開図を探す必要がある。

This sounds like we have added requirements and made our job harder, but in fact we have actually simplified things. One of the things that makes origami design so difficult is the interrelatedness of all the parts of the base; if you try to change the dimensions of one part of the model, you have to take into account how that part interacts with the rest of the paper. But it turns out that when you have a square broken up into active polygons, you can separate the design into independent pieces. Recall that the tree is the projection of the base, which is folded from the complete square. Each polygon on the square corresponds to a portion of the overall base and if you collapse any polygon into a section of the base — which I call a **subbase** — the projection of the subbase is itself a portion of the projection of the complete base, i.e., a portion of the original tree. For example, figure 5.8 shows the polygons for our six-legged base and the corresponding trees for each subbase. Note that since all of the corners of an active polygon must be

terminal nodes, the triangles at the bottom corners of the square are not active polygons and infact do *not* contribute to the base at all.

このことは、条件を追加して問題をより難しくするように思われるかもしれないが、実際には問題をより簡単にしている。折り紙の創作を困難にしている原因の一つは、基本形の全ての部分が相互に関連づけられていることである。もし作品のある部分の寸法を変えようとした場合、その部分が紙の他の場所にどのように影響を及ぼすのかということを考慮に入れなければならない。しかし、正方形を実効多角形に分割してしまえば、創作を独立した部分に分けることが出来ることが分かる。木が基本形の写像であり、その基本形は完全な正方形から折りたたまれていることを思い出してほしい。正方形上の各多角形は基本形全体の一部分に対応し、任意の多角形を基本形の一部（これを副基本形と呼ぶことにする）に折り畳んだ場合、その副基本形の写像はそれ自身が完全な基本形の写像の一部であり、したがって元の木の一部でもある。例えば、図 5.8 は、前述の 6 つのカドをもつ基本形における多角形と、それぞれの副基本形に対応する木を示している。なお、すべての実効多角形の頂点は端節点でなければならないため、正方形の下部の両端にある三角領域は実効多角形でなく、実は基本形に対して全く寄与していないことを付記しておく。

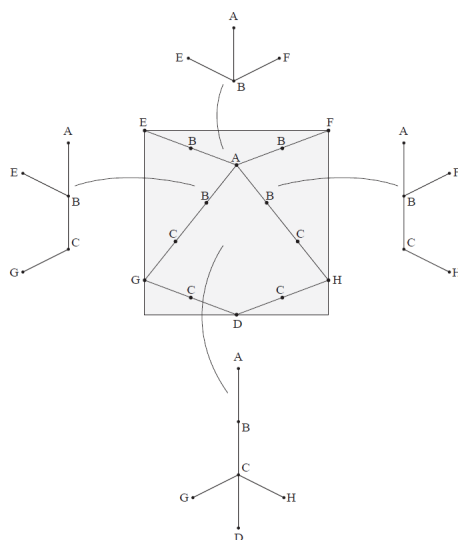


Figure 5.8. The four active polygons for the six-legged base and the trees corresponding to each subbase.

図 5.8.6 つのカドを持つ基本形における 4 つの実効多角形と、それぞれの副基本形に対応する木

An incredibly useful property of active polygons is that although two polygons may share a common side, if that side is an active path, any crease pattern that collapses one polygon into a subbase is *always* compatible with a crease pattern that collapses the adjacent polygon into its subbase! If you like, for example, you could cut up the square into separate polygons, fold each polygon into a subbase, and then glue matching edges of the subbases back together and be assured that they would match up. This matching property is unique to active paths, so any two polygons that share a side that is an active path will be guaranteed to match up. The practical upshot of this is that in order to find the creases that collapse the entire base, we can find crease patterns that collapse each polygon into a section of the overall base, taking the polygons one at a time. So here is a way to break up the design of a complicated base into a bunch of much simpler bases. To collapse the entire square into a uniaxial base, it suffices to find a way to collapse each polygon individually into the appropriate subbase.

実効多角形の非常に有用な特性は、二つの実効多角形が同じ辺を共有していたとしても、もしその辺が実効パスであれば、一つの多角形を副基本形に折り畳むような任意の展開図が、隣接する多角形を副基本形に折り畳むための展開図と常に共存できるということである。例えば、正方形を各多角形に切り分けて分割し、それぞれの多角形を副基本形に折り、そして副基本形同士で対応する辺を糊づけして元に戻そうと思ったときに、それぞれがうまく結合することが保証される。この結合特性は実効パスの優れた点であり、共有する辺が実効パ

スであるような任意の二つの多角形がうまく結合することを保証する。この特性の実用的な結果として、基本形全体を折り畳むための展開図を探す際に、一度に一つの多角形について、その多角形を基本形全体の一部に折り畳むような展開図を探せばよい。こうすることで、複雑な基本形の創作を、多くのずっと簡単な基本形の創作に分割することができる。正方形全体を一つの単軸基本形に折り畳むためには、それぞれの多角形を個別に適切な副基本形へと折り畳むための方法を探すだけで十分である。

Now, there is conceivably an infinite number of polygons and a polygon can have any number of sides, so we're not out of the woods yet. We need to find a way to collapse an arbitrary polygon into its corresponding subbase. Let's look at a few simple cases.

しかし、おそらく無限の種類の多角形が存在しており、また多角形の辺の数も様々であるため、まだ問題は解決されていない。任意の多角形に対応する副基本形へと折り畳むための方法を探す必要がある。いくつかの簡単な場合について考えてみよう。

We'll start with triangles. A triangle has three sides and three corners. Recall that the corners of active polygons are terminal nodes. Thus, the tree of a triangle must be a tree with three terminal nodes and vice-versa. There is only one graph with three terminal nodes, which is shown in figure 5.9. This tree corresponds to a uniaxial base with three flaps, whose lengths I have dubbed a , b , and c .

まず最初に三角形について考える。三角形には三つの辺と三つの頂点がある。すでに述べたように、実効多角形の頂点は端節点となる。したがって、三角形に対応する木は三つの端節点を備えていなければならない、逆に三つの端節点を持つ木は三角形に対応する。三つの端節点を持つグラフは、図 5.9 に示されるようなもの一種類しか無い。この木は三つのカドを備えた単軸基本形に対応している。そのカドの長さを a , b , c とする。

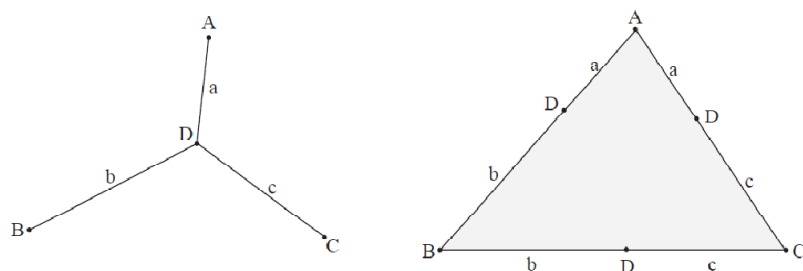


Figure 5.9. (Left) The tree with three terminal nodes, corresponding to a subbase with flaps of length a , b , and c . (Right) The corresponding triangle polygon.

図 5.9 (左) 長さが a , b , c のカドを持つ副基本形に対応する、三つの端節点を持った木。(右) それに対応する三角形。

Now, one of the assumptions we've made is that the sides of the active polygons are active paths. For a triangle whose three sides are active paths, that means we already know the lengths of the sides: they are, respectively, $(a+b)$, $(b+c)$, and $(c+a)$. A triangle is uniquely specified by its sides, so if we are given the tree we know what the triangle is and vice versa. Given a triangle with sides defined above, is there a crease pattern that collapses this triangle into a subbase with flaps of the appropriate length?

ここで、実効多角形の各辺は実行パスであると仮定されていた。三角形の三つの辺が実効パスであるため、その辺の長さが確定する。すなわち、それぞれ $(a+b)$, $(b+c)$, $(c+a)$ となる。三角形は三辺の長さによって一意に特定されるため、木が与えられればその三角形を得ることができ、逆に三角形を与えられればその木を得ることができる。上で定義された辺を持つ三角形が与えられたとき、その三角形を適切な長さのカドを備えた副基本形に折り畳むことのできる展開図は存在するだろうか？

There is such a crease pattern and it is one familiar to every paper folder: the humble rabbit ear. It's been known since Euclid that the three bisectors of a triangle intersect at a point, which enables one to form a rabbit ear from any triangle with all of the edges lying in the same plane (or if you flatten the rabbit ear, the edges all lie on a line). One can also show the converse: if three sides of a triangle are $(a+b)$, $(b+c)$, and

$(c+a)$, then the length of the flaps of the resulting rabbit ear are a , b , and c . So any triangle bounded by three active paths can be filled in with the creases of a rabbit ear, as shown in figure 5.10.

そのような展開図は存在しており、それはあらゆる折り紙愛好家がよく知っているものである。すなわちそれは単純なつまみ折りである。ユークリッド以来、三角形の三つの角の二等分線が一点で交わることが知られており、それによって任意の三角形をつまみ折りすれば、すべての辺を同一平面上に乗せることができる（または、つまみ折りを平らにすれば、すべての辺は同一直線上に乗る）。また、逆を示すことも可能である。つまり、もし三角形の三辺が $(a+b)$ 、 $(b+c)$ 、 $(c+a)$ であれば、それをつまみ折りしたときのカドの長さは a 、 b 、 c となる。したがって、三つの実効パスで囲まれた任意の三角形は、図 5.10 に示すようなつまみ折りの展開図で中を埋めることができる。

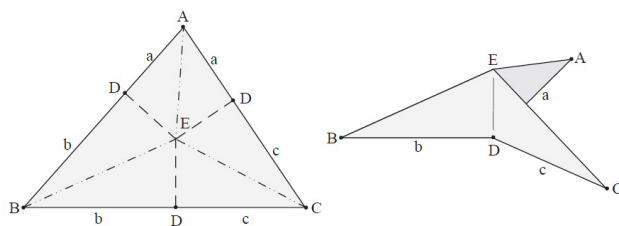


Figure 5.10. Crease pattern for a rabbit ear and resulting subbase. Note that the three flaps of the subbase have the same lengths as the three edges of the tree.

図 5.10 つまみ折りの展開図とそれを折り畳んだ副基本形。副展開図の三つのカドは、木の三本の辺と同じ長さであることに注意。

If you are actually folding the active polygon, you can find the intersection of the angle bisectors— point E in the figure — just by pinching each corner in half along the bisector and finding the point where all the creases come together. If you are calculating the crease locations numerically, there is a beautiful formula for the location of the intersection of the angle bisectors of an arbitrary triangle. If \mathbf{p}_A , \mathbf{p}_B , and \mathbf{p}_C are the vector coordinates of corners A , B , and C and \mathbf{p}_E is the coordinate of the bisector intersection, then \mathbf{p}_E is given by the simple formula

実効多角形を実際に折り畳む場合、各頂点を角の二等分線に沿って半分に折るように指でつまみ、すべての折れ線が集まる場所を探すだけで、角の二等分線の交点（図中の点 E ）を見つけることができる。計算で折り目の位置を求める場合、任意の三角形の角の二等分線の交点の位置を求める美しい公式が存在する。頂点 A 、 B 、 C の位置ベクトルを \mathbf{p}_A 、 \mathbf{p}_B 、 \mathbf{p}_C とし、 \mathbf{p}_E を二等分線の交点の座標とした場合、 \mathbf{p}_E は次の単純な式で求められる。

That is, the location of the bisector intersection is simply the average of the coordinates of the three corners with each corner weighted by the length of the opposite side.

すなわち、二等分線の交点の位置は各頂点の座標に反対側の辺の長さを掛け合わせたものの単純な平均値である。

What happens when one of the sides of the triangle is not an active path? This can happen, for example, when one of the sides of the triangle lies along an edge of the square; all of the triangles in figure 5.8 are of this type. Since the distance between any two terminal nodes must be greater than or equal to the minimum path length, the side that isn't an active path must be slightly too long to be an active path rather than too short. Fortunately, only a slight modification of the rabbit ear is necessary to address this situation. Figure 5.11 shows the crease pattern and subbase when side BC is slightly too long.

三角形の辺の一つが実効パスでない場合はどうなるだろうか？これは例えば、三角形の辺の一つが正方形の輪郭と重なっていた場合に発生する。図 5.8 中の三角形はすべてこの種類のものである。任意の二つの端節点の間の距離はパスの最小長以上でなければならないため、実効パスではない辺は、実効パスより短いことはなく、むしろ常に長くなる。この状況には、幸いにも、つまみ折りに少しの変更を加えるだけで対応することができ

る。図 5.11 は辺 BC がわずかに長すぎる場合の展開図と副基本形を示している。

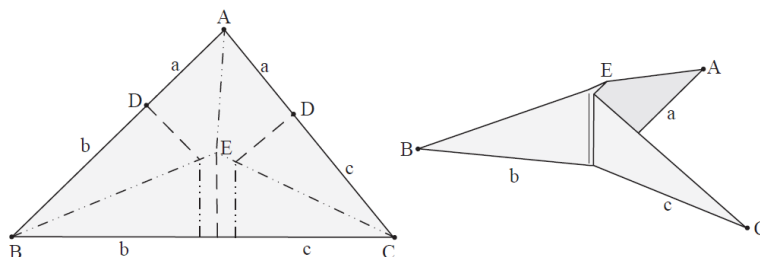


Figure 5.11. (Left) Crease pattern for a triangle when side BC is not an active path. (Right) Resulting subbase.

図 5.11 三角形の辺 BC が実効パスではない場合の展開図（左）とそれを折った副基本形（右）

If the triangle has two sides that aren't active paths, a similar modification will still collapse it appropriately.

三角形が実行パスでない辺を二つ持っていた場合も、同様の変更で適切にそれを折りたたむことができる。

Another case that we should consider is a triangle graph that has one or more internal nodes along its edges, which may or may not be due to a kink in the subgraph. For example, the two side subgraphs in figure 5.8 each have three terminal nodes, but in each graph, one of the edges has an internal node because the subgraph has a kink at that point. We can still use the rabbit ear construction to find most of the creases, but wherever we have an internal node along an active path, we need a crease radiating inward from the internal node to the mountain fold that forms the “spine” of the subbase.

もう一つ考慮すべき点は、三角形に対応する副グラフに折れ曲がりがあるとき、三角形の辺上に一つ以上の内節点が存在する場合があるということである。例えば、図 5.8 中の左右にある副グラフはそれぞれ三つの端節点を持つが、それぞれ副グラフの一边に折れ曲がりがあるため、その点に内節点を持つ。その場合でもつまみ折り構造を用いることでほとんどの折り目を見つけることが可能であるが、実行パス上に内節点がある場所では、その内節点から副基本形の「尾根」を構成する山折り線まで内部へ向けて折り目を伸ばす必要がある。

We might wonder which creases are valley and mountain folds. All of the creases that form the spine of the subbase are always mountain folds. Most of the creases corresponding to active paths are valley folds; however, for terminal nodes that fall in the interior of the paper, there is a mathematical requirement that the number of mountain and valley folds that come together at a point differ by 2; consequently, for any terminal node that lies in the interior of the square, one of the active paths must be converted to a mountain fold.

それぞれの折り線が谷折りになるか山折りになるかを考えてみよう。副基本形の尾根を構成する折り線は常に山折りとなる。実効パスに対応する折り線のほとんどは谷折りであるが、紙の内部に位置する端節点については、同じ点に集まる山折りと谷折りの数の差が常に 2 になるという数学的な条件が存在しているため、正方形の内部にあるすべての端節点について、実効パスの一つを山折りに変更しなければならない。

The other type of creases we will encounter are those creases that radiate inward from internal nodes; they can be mountain folds, valley folds, or no folds at all, depending on which way we lay the flaps when we flatten the uniaxial base. Since any such fold has three possible states, I call them **tri-state** folds. To give some indication which way the folds go, I'll draw all of the active paths as valley folds (even though a few of them may actually turn out to be mountain folds) and draw all of the tri-state folds as x-ray lines (since they can go either way).

他に出現する折り線の種類としては、内節点から内側へ伸びているような折り線がある。それらは、単軸基本形を折りたたむ際にカドをどちらに倒すかによって、山折り、谷折り、またはまったく折らない場合のいずれかになりうる。そういった折り線は三つの状態を取りうるため、三状態の折り線と呼ぶこととする。折り線の

向きをおおまかに示すために、すべての実効パスを（たとえいくつかは実際には山折りであったとしても）谷折りとして描き、すべての三状態の折り線は（どちらの折り方にもなりうるため）透視線として描くことにする。

Using the rabbit-ear crease pattern, we can fill in three of the four active polygons of the six legged base as shown in figure 5.12. Using similar patterns, we can fill in the crease patterns for any triangle in any network of active polygons. Let us now turn our attention to the next case: four-sided active polygons.

つまみ折りの折り線パターンを用いることで、6つのカドをもつ基本形中の4つの実効多角形のうちの3つを、図5.12のような折り線で埋めることができる。同様のパターンを用いることで、任意の実効多角形の集合中の任意の三角形についても折り線で埋めることが可能である。そこで次に、4つの辺をもつ実効多角形について考えてみる。

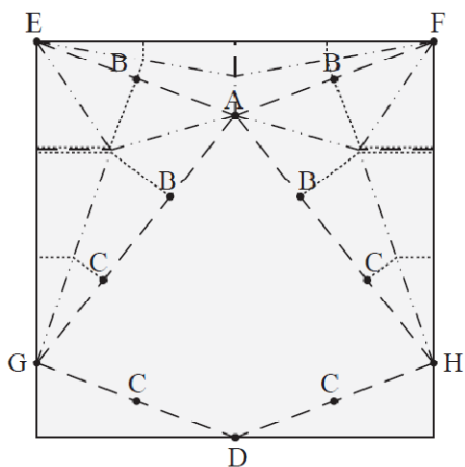


Figure 5.12. Crease pattern for the six-legged base with triangles filled in.

図 5.12 三角形を折り線で埋めた、6つのカドを持つ基本形の展開図

Just as a triangular polygon corresponded to a tree with three terminal nodes, a four-sided polygon, or quadrilateral, has a tree with four terminal nodes. There are two distinct types of trees with four terminal nodes, shown in figure 5.13. I call these two graphs the “4-star” and the “sawhorse.” If a , b , c , and d are the lengths of the four terminal edges in either graph and e is the length of the “body” in the sawhorse, then a quadrilateral whose sides are all active paths will have sides of length $(a+b)$, $(b+c)$, $(c+d)$, and $(d+a)$ for the 4-star and $(a+b)$, $(b+c+e)$, $(c+d)$, and $(d+a+e)$ for the sawhorse. There is an ambiguity associated with quadrilateral polygons; with the triangle, if we are given the lengths of the three sides, then the lengths of the edges of the tree are uniquely determined but with a quadrilateral, knowledge of just the lengths of the sides is insufficient to fully specify the dimensions of the tree. We must have the full tree to know whether the graph is a 4-star or a sawhorse and in the latter case, to know the length of the body.

三角形が三つの端節点をもつ木に対応するように、四つの辺をもつ多角形すなわち四角形は、四つの端節点を備えた木に対応する。四つの端節点をもった木には、図5.13に示すように二つの異なる種類が存在する。これらの二つのグラフをそれぞれ「四角星」と「ノコギリ台」と呼ぶことにする。 a , b , c , d をそれぞれのグラフの四つの終端辺の長さとし、 e をノコギリ台の「台」の長さとした場合、すべての辺が実効パスであるような四角形の辺の長さは、四角星の場合は $(a+b)$, $(b+c)$, $(c+d)$, $(d+a)$ 、ノコギリ台の場合は $(a+b)$, $(b+c+e)$, $(c+d)$, $(d+a+e)$ となる。四角形については曖昧さが存在している。すなわち、三角形の場合は三辺の長さが与えられれば木の辺の長さが一意に決まるのに対し、四角形の場合は辺の長さに関する情報だけでは木を決定するのに十分ではないということである。グラフが四角星かノコギリ台かを知るには、そして後者の場合には台の長さを知るには、完全な木が必要である。

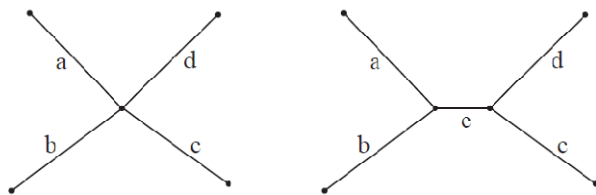


Figure 5.13. The two trees with four terminal nodes. (Left) The 4-star. (Right) The sawhorse.

図 5.13 4 つの端節点を備えた二つの木。(左) 四角星 (右) ノコギリ台

Finding crease patterns for quadrilaterals is more complex than finding patterns triangles because there is more than one crease pattern that will collapse a given quadrilateral into the appropriate subbase. In fact, there can be infinitely many different crease patterns for a given quad. However, there is one pattern that has the desirable property that it works for quadrilaterals corresponding to either type of tree. I call it “the gusset quad” crease pattern because it usually has a gusset across its top. Its construction is rather complex, but I will describe it briefly here.

四辺形の展開図を見つけることは、三角形の展開図を見つけることよりもより複雑である。なぜなら、与えられた四辺形を適切な副基本形に折り畳める展開図が複数存在するためである。実際、与えられた四角形について無限に多くの異なる展開図が存在することがある。しかしながら、どちらのタイプの木に対応する四辺形にでも適用できる、望ましい特性をもったパターンが一つある。それを折ると通常頂上に三角形のマチができるため、このパターンを「マチ付き四辺形」と呼ぶことにする。その構築は多少複雑であるが、簡潔に説明する。

Given a quadrilateral $ABCD$ as shown in figure 5.14. Inscribe a smaller quadrilateral inside whose sides are parallel to the sides of the original quadrilateral but are shifted inward a distance h (the value of h is not yet determined). Denote the corners of the new quadrilateral by A' , B' , C' , and D' . Drop perpendiculars from these four corners to the sides of the original quadrilateral. Label their points of intersection A_{AB} where the line from A hits side AB , B_{AB} where the line from B hits AB , and so forth.

図 5.14 に示されるような四辺形 $ABCD$ が与えられたとする。その中に小さな四辺形を置き、それぞれの辺が元の四辺形の辺と平行で距離 h (h の値はまだ決定しないでおく) だけ内側にずらされているようにする。新しい四辺形の頂点を A' , B' , C' , D' とする。そしてそれら四つの頂点から元の四辺形の辺へ垂線をおろす。それぞれの交点を、 A からおろした垂線と AB との交点を A_{AB} , B からおろした垂線と AB との交点を B_{AB} というように名づける。

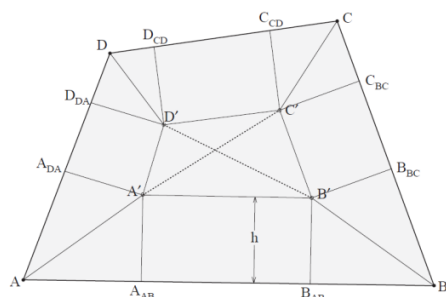


Figure 5.14. Construction of the gusset quad for a quadrilateral $ABCD$. Inset the quadrilateral a distance h ; then drop perpendiculars from the new corners to the original sides.

図 5.14 四辺形 $ABCD$ におけるマチ付き四辺形の作図。距離 h だけ内側に四辺形を挿入し、新しい頂点から元の辺に垂線を引く。

Now we need some distances from the tree. Let l_{AC} be the distance from node A to node C on the tree and l_{BD} be the distance from node B to node D . In most cases (see below for the exceptions), there is a unique solution for the distance h for which one of these two equations holds:

ここで、木から得られる距離が必要となる。木上の節点 A から節点 C までの距離を l_{AC} とし、節点 B から節点 D までの距離を l_{BD} とする。ほとんどの場合（例外については下を参照）、次の二つの方程式のいずれかを満たすような距離 h の解がただ一つ存在する。

Let us suppose we found a solution for the first equation. The diagonal $A'C$ divides the inner quadrilateral into two triangles. Find the intersections of the bisectors of each triangle and call them B'' and D'' . (If the second equation gave the solution, you'd use the opposite diagonal of the inner quadrilateral and find bisector intersections A'' and C'' .) The points A' , B'' , C' , and D'' are used to construct the complete crease pattern.

最初の方程式の解が見つかったとしよう。対角線 $A'C$ は内側の四角形を二つの三角形に分割する。それぞれの三角形で角の二等分線の交点を探し、それらを B'' と D'' とおく（もし二番目の方程式に解が存在した場合、内側の四角形のもう一方の対角線を用い、角の二等分線の交点 A'' と C'' を求める）。四つの点 A' 、 B'' 、 C' 、 D'' を使うと、完全な展開図を作図することができる。

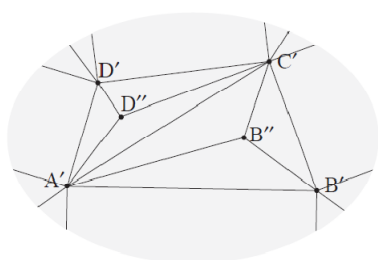


Figure 15. On the inner quadrilateral, construct the bisectors of each triangle to find points B'' and D'' .

図 15 内側の四角形で、それぞれの三角形の角の二等分線の交点を作図し、 B'' と D'' を求める。

You've now found all of the points necessary to construct the crease pattern; connect them with creases as shown in figure 16 to produce the mountain folds that form the spine and the one valley fold that forms the gusset. You will also have to construct tri-state creases from each internal node along the sides of the quad to complete the crease pattern. Figure 5.16 also shows the folded gusset quad and its tree.

これで、展開図を作図するために必要なすべての点が見つかった。これらの点を図 16 に示すような折り線でつなぐと、尾根となる何本かの山折り線と、マチを折るための一本の谷折り線が得られる。展開図を完成させるためには、四角形の辺上のそれぞれの内節点から三状態の折り線を引く必要がある。図 5.16 は折り畳んだマチ付き四角形とその展開図を示す。

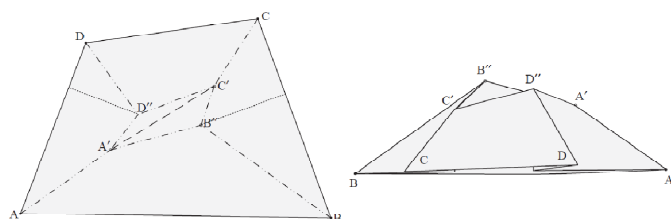


Figure 5.16. (Left) The mountain and valley folds of the gusset quad. Two tri-state creases are shown. (Right) The folded version of the gusset quad.

図 5.16（左）マチ付き四角形の山折り線と谷折り線。二本の三状態の折り線も示す。（右）マチ付き四角形を折り畳んだ様子。

You can construct an equation for the distance h in terms of the coordinates of the four corners and the distances; it's a rather involved quadratic equation, but can be solved using a pocket calculator and high-school algebra. For arbitrary quadrilaterals, there is not a simple method to find the crease pattern by

folding, but symmetric quadrilaterals (such as the one in the middle of figure 5.8) can be found by folding alone.

4つの頂点座標とそれらの距離から、距離 h をもとめる方程式を作ることができる。それはかなり複雑な二次方程式であるが、ポケット電卓と高校の代数を使って求めることが可能である。任意の四辺形について、折り畳むことで展開図を見つける単純な方法は存在しないが、対照的な四辺形（図 5.8 の中央のような）であれば折り畳むだけで見つけることが可能である。

I alluded to exceptions above; there are quadrilaterals for which the points A' , B' , C' , and D' all fall on a line. In these special cases, you don't get an inner quadrilateral; all of the inner creases collapse onto a line (or sometimes a point) and you get the simplified crease pattern shown in figure 5.17. Jun Maekawa has proven a theorem — called, appropriately enough, the Maekawa theorem — that says, in essence, any quadrilateral can have its edges collapsed onto a line using a crease pattern similar to this one. However, only in a relatively small number of cases will the lengths of the resulting flaps match the desired tree.

すでに示唆したように、これには例外があり、点 A' , B' , C' , D' が一直線上にあるような四角形が存在する。このような特別な場合では内部の四辺形が見られず、内部の全ての折り線がひとつの線（場合によっては点）へと押し潰され、図 5.17 のようなより単純な展開図となる。前川淳は、適切にも前川定理と呼ばれる定理を証明したが、それは本質的に、いかなる四辺形についても、この展開図と似たような折り方で、辺が一直線に並ぶように折りたたむことができると述べている。しかしながら、得られた基本形のカドの長さが希望する木と一致することは滅多にないだろう。

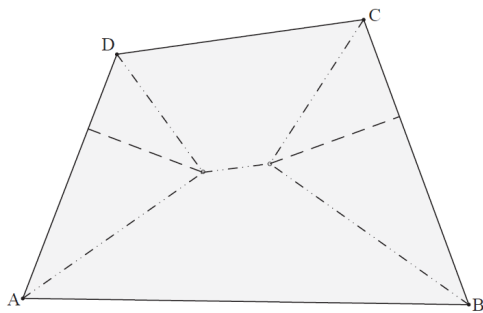


Figure 5.17. The Maekawa crease pattern.

図 5.17 前川分子の展開図

The Japanese folder Toshiyuki Meguro has extensively explored crease patterns that collapse polygons onto lines and has coined the name “bun-shi,” or “molecules,” to describe such patterns. Just as individual molecules fit together to make a larger biological structure, so too do origami molecules fit together to make an origamical structure: the base.

日本の折り紙作家である目黒俊幸は、多角形を一直線上に折りたたむ展開図を広く調査し、そのようなパターンを説明するための「分子」という名前を考案した。より大きな生物学的構造を作るために個々の分子をお互いにつなげ合うように、折り紙的構造を作るために折り紙分子をお互いにつなげ合うのである。それがすなわち、基本形である。

Using the rabbit ear crease pattern for triangles and the gusset quad crease pattern for quadrilaterals, you can fill in any active polygon network that consists of triangles and quadrilaterals to get the complete crease pattern for the base. Such a polygon network is the one for the six-legged base considered earlier in this article. Figure 5.18 shows the full crease pattern for the six-legged base and the resulting base. Unlike figure 5.1, this base is a real, foldable base; you can easily verify the crease pattern by cutting it out and folding it on the lines. As you can see, the projection of the base into the plane is indeed the tree, and all of the flaps have their proper length.

つまみ折りを使った三角形の展開図やマチつき四辺形の展開図を使うことで、三角形や四辺形で構成されてい

るような任意の実効多角形群をうめることができ、基本形のための完全な展開図を得ることができる。そのような多角形群のひとつが、この章ですでに取り上げた6つのカドを持つ基本形である。図5.18は6つのカドを持つ基本形の完全な展開図と、折り畳んだ結果としての基本形を示している。図5.1と異なり、この基本形は実際の、折り上げることが可能な基本形である。線に沿って切り抜いて折ってみれば、簡単に展開図を検証することができる。そうすれば分かるように、この基本形を平面へ射影したものは正に必要な木であり、全ての基本形のカドはそれぞれ適切な長さを持っている。

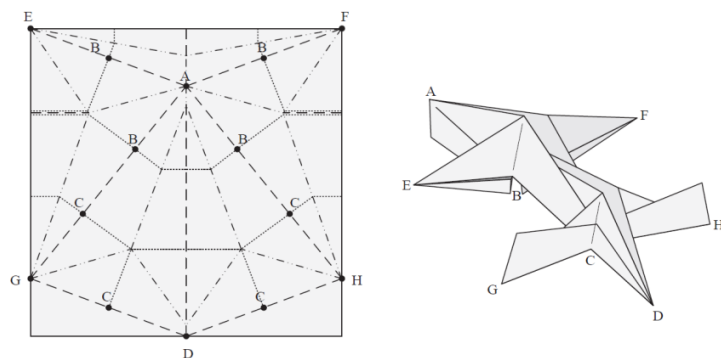


Figure 5.18. Full crease pattern and six-legged base.

図 5.18 完全な展開図と6つのカドを持つ基本形

If you try to collapse figure 5.18 into the base, you will have to flatten some valley folds and turn several tri-state creases into valley or mountain folds, depending on how you stack the layers and arrange the points. Although Tom Hull, Toshikazu Kawasaki, and others have identified several rules for assigning mountain and valley folds to a crease pattern that will allow it to be folded flat, I haven't yet identified an algorithm to assign mountain and valley folds to a tree method crease pattern — in fact, as you can tell by folding up a base, there is always more than one distribution of mountain and valley folds for a given crease pattern. In any event, when you collapse the base, all of the points will be free and unattached from the others and each segment of the base is precisely the same length as corresponding segment on the tree. You can thin the points further and add reverse folds, et cetera, to turn the base into a subject.

図5.18を基本形へ折り畳もうとした場合、いくつかの谷折りを平らにしなければならず、いくつかの三状態の折り目を谷折りか山折りにしなければならないだろう。それはどのように面を重ねたり、カドをまとめたりするかに依存する。Tom Hullと川崎敏和、その他によって、展開図を平らに折り畳むことが可能になるような山折り・谷折りの配置の規則が特定されているが、私はまだTree Methodにおける展開図の山折り・谷折りの配置のアルゴリズムについて特定を行っていない——実際、基本形を折ってみれば分かるように、与えられた展開図については常に複数の山折り・谷折りの配置パターンが存在しているのである。いずれにせよ、基本形を折りたためば、すべてのカドは束縛されておらず他から独立しており、基本形のそれぞれの部分是对应する木上の部分と正確に同じ長さである。カドを細くしたり中割折りを加えたりすることによって、基本形を対象の形状へ変形させることが出来る。

Much the same procedure can be used for any network of active polygons. However, what happens if there are polygons with five, six, or more sides? You saw the jump in complexity going from three to four corners was considerable. Although there was only one type of tree for a triangle and two for a quadrilateral, for a five-sided polygon there are three possible types of trees and the number rises quickly beyond that. So there are many more possibilities to enumerate. In addition, computation of the crease pattern for higher-order polygons gets very complicated very quickly, and you can imagine the difficulties as the number of points increases. For a nineteen-pointed insect, the network of active polygons could conceivably consist of a single nineteen-sided polygon! How would we ever collapse such a beast?

ほとんど同じ手法を任意の実効多角形群に適用させることができる。しかし、五角形、六角形、あるいはそれ以上の多角形がある場合はどうだろうか？すでに見たように、三角形と四角形の間の複雑性の飛躍は相当なものである。三角形には一種類の木しかなく、四角形には二種類しかないが、五角形は三種類の可能な木が存在

し、六角形以上になるとその個数は急速に増大していく。そのため列挙すべき状態がはるかに多数存在する。加えて、高次多角形の展開図の計算は急速に複雑になるので、頂点が増えた時の複雑さを想像することができる。19 のカドがある昆虫について、実効多角形群が一つの 19 角形だけからなるということも考えられる。そういった化け物をどうやって折り畳めばいいのだろうか？

I recently discovered one solution: there is a generalization of the gusset quad that produces a crease pattern for any active polygon. However, its construction is even more complex than the construction of the gusset quad was. It turns out, though, that we need no more than the gusset quad and the rabbit ear to fold a base for any tree.

私は最近、ある解決策を発見した。任意の実効多角形の展開図を生成できるような、マチ付き四辺形の一般形が存在するのである。しかし、その構築はマチ付き四角形の構築よりもさらに複雑である。もっとも、次に見るように、任意の木に対応する基本形を折るためにはマチ付き四辺形とつまみ折り以上のものは必要ない。

Again, a paradox: we'll simplify the design problem by making the tree more complicated. Consider the tree in figure 5.19, which is a 5-pointed star. This graph leads to a set of active paths that comprise a single five-sided polygon.

ここで再び矛盾したことを考える。すなわち、木をより複雑化することで、創作上の問題を単純化する。図 5.19 の五角星の木について考えてみよう。このグラフからは、一つの五角形を構成する実効パス群が導かれる。

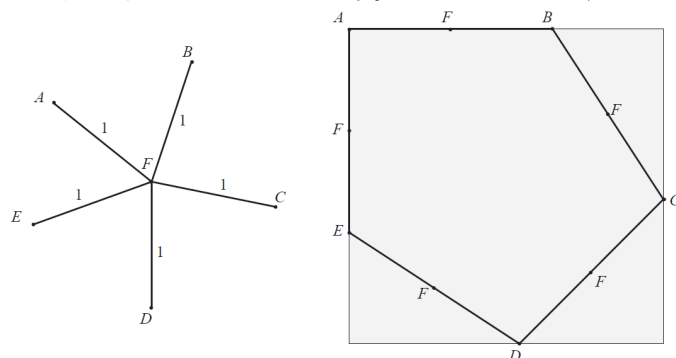


Figure 5.19. (Left) Tree for a base with 5 equal flaps. (Right) Pattern of terminal nodes and active paths corresponding to this tree.

図 5.19 5 つの等しい長さのカドをもつ基本形の木(左)。この木に対応する端節点と実効パスのパターン (右)

Now suppose we wanted to add one more point to the star. That would entail adding one more terminal node — node G — to the active polygon network. On the tree graph, the new node would be connected to node F as shown in figure 5.20. (If we connected it to one of the other nodes that wouldn't add another point; it would just lengthen an existing point.) Suppose the point has a length l . Then since all the other edges are 1 unit long, the tree theorem tells us that the new terminal node must be separated from each of the other terminal nodes by a distance $(1+l)$.

ここで、この星にもう一つのカドを加えたいと思ったとしよう。その場合、この実効多角形群にもう一つの端節点 — 端節点 G — を追加することになる。木グラフ上では、新しい節点は図 5.20 のように節点 F と接続される (もし他の節点に接続したら、それは別のカドを加えたことにならない。既存のカドの長さを伸ばすだけである)。新しいカドの長さを l とする。他のすべての辺が一単位長であるため、ツリー定理により、新しい端節点は他の端節点から距離 $(1+l)$ だけ離されなければならない。

We can visualize these constraints by imagining that each terminal node is surrounded by a circle whose radius is equal to the length of the edge attached to that node in the tree. That is, nodes $A-E$ are surrounded by circles of unit radius, while node G is surrounded by a circle of radius l . The requirement that G be separated from the other nodes by at least a distance $(1+l)$ is equivalent to the requirement that circle G not overlap with any other circle.

この制約は、それぞれの端節点が、木においてその節点到に接続されている辺の長さに等しい半径の円で囲まれているという形で可視化することができる。つまり、節点 $A-E$ は単位円で囲まれていて、一方、節点 G は半径 l の円で囲まれているということである。 G が他の節点から少なくとも $(l+1)$ の距離だけ離されているという条件は、円 G が他のどの円にも重なっていないという条件と同値である。

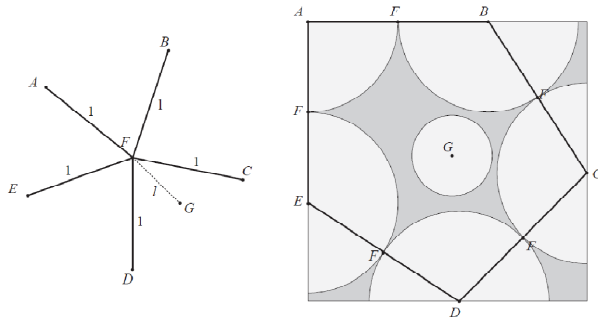


Figure 5.20. (Left) Tree with a new terminal node. (Right) One possible position for node G that satisfies the tree theorem.

図 5.20 新しい端節点のある木 (左)、ツリー定理を満たす、節点 G の可能な位置の一つ (右)。

Figure 5.20 shows the circles around each node. As long as the circles do not overlap, the tree theorem is satisfied. As l , the length of the new edge, is increased, the size of the circle around node G must be increased as well. Eventually, circle G will swell until it is touching at least three other circles, and at that point, shown in figure 5.21, the new point is as large as it can possibly be.

図 5.20 はそれぞれの節点を囲む円を表している。円が重ならない限りツリー定理は満たされている。新しい辺の長さである l が大きくなるに従って、節点 G を囲む円も同様に大きくななければならない。最終的に円 G は図 5.21 のように他の三つの円に接触するまで大きくなり、その時、新たなカドはとりうる限りで最大の大きさとなる。

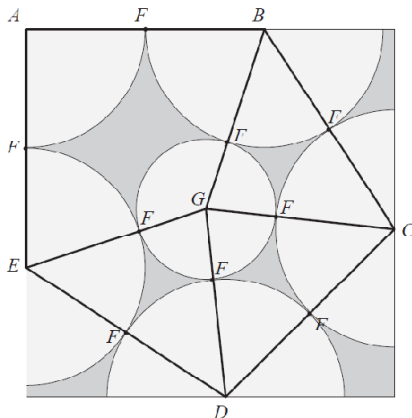


Figure 5.21. Terminal node pattern with the largest possible circle around node G .

図 5.21 節点 G において、とりうる最大の大きさの円を持つ端節点のパターン

Wherever circle G touches another circle, the two terminal nodes are spaced at their minimum separation. Consequently, the paths between the nodes of touching circles are active paths. Whenever we add a new node inside a polygon, if we make the corresponding flap sufficiently large, it forms several new active paths with the vertices of the old active polygon. In the process, it breaks up the polygon into smaller polygons with fewer sides than the original polygon. In figure 5.21, adding the new point has broken the

five-sided polygon into one quadrilateral and three triangles. But we already know crease patterns for quadrilaterals and triangles! Filling in the crease patterns in each polygon gives a crease pattern for the full base, shown in figure 5.22.

円 G が他の円とどこで接触したかにかかわらず、円に対応する二つの端節点の間隔は最小長となる。その結果、接触した円に対応する節点間のパスは実効パスとなる。多角形の中に新しい節点を追加する際、それに対応するカドの大きさを十分に大きくすれば、古い実効多角形の頂点との間にできた新しいパスは常に実効パスとなる。その過程で、多角形がより小さい多角形群に分割され、それらの小多角形は元の多角形よりも頂点の数が少なくなる。図 5.21 では、新しい点を追加することで五角形が一つの四辺形と三つの三角形に分割されている。そして、四辺形と三角形の展開図はすでに判明しているのである！それぞれの多角形を展開図で埋めることで、図 5.22 に示すような完全な基本形の展開図を得ることができる。

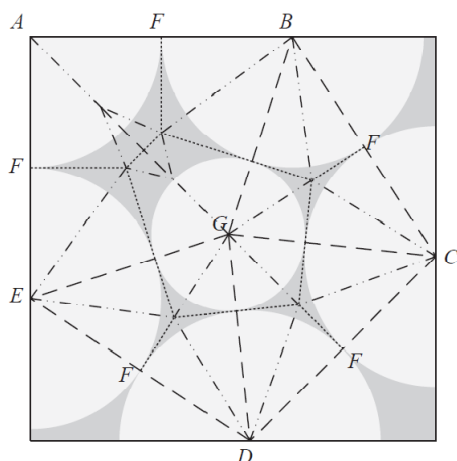


Figure 5.22. Crease pattern for the 5-pointed base.

図 5.22 五角形の基本形の展開図

Imagining terminal nodes as being surrounded by circles makes it easier to visualize the path constraints of the tree theorem. Although strictly speaking, the circle analogy only holds for terminal nodes with exactly one internal node between them, one can devise a similar condition for more widely separated nodes. If you draw in circular arcs around each terminal node tangent to the tri-state creases, you get a set of contours on both the crease pattern and the base, in which edges attached to terminal nodes are represented by circles and edges attached only to internal nodes are represented by contours that snake through the crease pattern as in figure 5.23. For numerical computation, I find that it is simpler to do all calculations in terms of paths and path lengths, but for intuition and visualization, the circles and contours constitute an equivalent and more easily visualized tool for devising new crease patterns.

端節点が円に囲まれていると考えると、ツリー定理におけるパスの制約を容易に可視化することができる。しかし厳密に言えば、この円の考え方は複数の端節点の間に内節点の一つだけしかないという状況のみにしか当てはまらないが、間に複数の節点がある場合にも似たような状況を工夫することができる。端節点を中心とした円弧を三状態の折り目に接するように描くと、展開図と基本形の両方に何本かの帯が得られる。端節点に接続した辺は円として表されるが、内節点のみに接続する辺は図 5.23 に示すように展開図の中を蛇のようにうねる帯として表される。数値計算に関しては、パスとパスの間の長さをすべて計算するほうが簡単だと思うが、直観的で視覚的な方法としては、円と帯は新しい展開図を考えるための同等かつ視覚的にわかりやすいツールとなる。

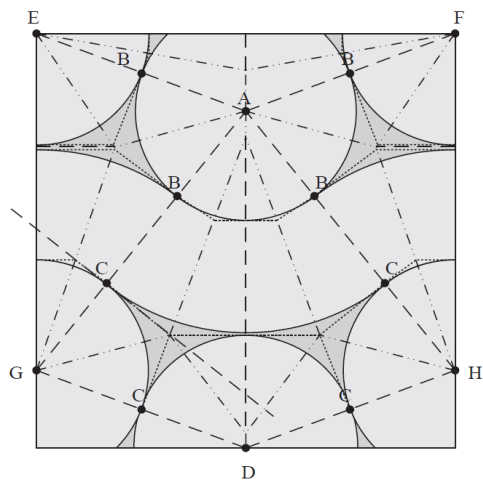


Figure 5.23. Circles and contours for the six-pointed lizard base.

図 5.23 6 つのカドを持ったトカゲの基本形に対応する円と帯

If you've stuck with me so far, you're now at the goal: an algorithm for finding a crease pattern to fold a base with any number of points from a square, rectangle, or any other shape paper. (Infact, this algorithm even works for circular paper!) To summarize the algorithm:

これまでのことを忠実に実行したら、目標が達成されたことになる。すなわち、正方形や長方形、あるいは他の任意の形状の紙から任意の数のカドを備えた基本形を折るための展開図を求めるアルゴリズムが得られた。(実際、このアルゴリズムは円形の紙にすら有効である！)そのアルゴリズムをまとめると次のとおりである。

1. Draw a tree, or stick figure, of the base, labeling each edge with its desired length.
基本形について、木、すなわち棒線画を描き、それぞれの辺に必要な長さを割り当てる。
2. Find a pattern of terminal nodes on the square that satisfies the tree theorem, namely, that the distance between any two nodes on the square is greater than or equal to their separation on the tree.
正方形上で、ツリー定理を満たすような端節点の配置を探す。つまり、任意の二つの端節点間の正方形上の距離を、対応する木上の距離以上にする。
3. Mark all of the active paths, paths whose actual length is equal to their minimum length. Identify the active polygons.
すべての実効パス、すなわち実測長が最小長と等しいパスを抽出する。それによって実効多角形を求める。
4. For any active polygon with five or more sides, add a node and edge to the tree attached at an internal node of the polygon and make the edge as large as possible, thereby breaking up the active polygon into quads and triangles.
五つ以上の辺をもったすべての実効多角形について、多角形の内部節点に接続する節点と辺を木に追加する。そして辺の長さをできるだけ長くし、実効多角形を四角形と三角形に分割する。
5. Fill in each active polygon with quad and triangle crease patterns.
それぞれの実効多角形の中を四角形と三角形の展開図で満たす。

The good news is that using the algorithm described above, a base can be constructed for *any* tree—in fact, there are usually *many* distinctly different solutions for a single tree. The bad news is, as you might suspect from some of the above, constructing the crease pattern can be computationally intensive.

上記のアルゴリズムを使うことで任意の木の基本形を求めることができるというのはよいことである。実際、一つの木に対してしばしばたくさんの明確に異なる解が存在する。しかし悪いことに、これまでの記述の端々から想像できるように、展開図を求めることは、数値計算的に困難である。

(The worse news is that even when you have the crease pattern, folding it up into a base can be infuriatingly difficult; there is rarely a step-by-step folding sequence. More often than not, you need to precrease everything, then collapse the base all at once.)

(さらに悪いことに、たとえ展開図ができたとしても、それを折り上げて基本形にすることがひどく困難なことがある。段階的な折り手順が存在することはまれで、あらかじめ折り目を付けておいて、一度で基本形へ折り畳む必要があることの方が多い。)

Since every polygon network can be broken up into triangles and quads by the addition of extracircles, the triangle and quad molecules are by themselves sufficient for filling in the crease pattern for any tree. However, there are many other possible molecules, including molecules that can be used for higher-order polygons. It turns out that the gusset quad is just a special case of a more general construction that is applicable to any higher-order polygon. I call this construction the **universal molecule**. In fact, many of the known molecules — the Maekawa and Meguro molecules, the rabbit ear, and so forth (but not the arrowhead quad, as it turns out) are special cases of the universal molecule. The rest of this article describes the construction of this molecule for an arbitrary polygon.

新たな円を追加することで任意の多角形群を複数の三角形と四角形に分割することができるため、任意の木の展開図を折り目で埋めるためには三角形と四角形の分子があれば十分である。しかし他にも多くの可能な分子が存在しており、その中には、より高次の多角形に用いることができるものがある。実際、マチ付き四辺形は、任意の高次多角形に対して用いることができるより普遍的な構造の、特殊な場合でしかない。そのような普遍的な構造を普遍的分子と名付ける。実は、前川分子や目黒分子、つまみ折りなどの多くの既知の分子（ただし四角矢じりは除く）は普遍的分子の特殊例である。この文書の残りで、任意の多角形に対する分子の構築について述べる。

Consider a general polygon that satisfies the tree theorem, i.e., any two vertices are separated by a distance greater than or equal to their separation on the tree graph. Since we are considering a single active polygon, we know that of the paths between nonadjacent vertices, none are at their minimum length (otherwise it would be an active path and the polygon would have been split).

ツリー定理を満たす一般の多角形を考える。すなわち、任意の二頂点間の距離は木グラフの中での距離以上になっている。ここでは単独の実効多角形について考えているので、隣接していない頂点間のパスについては、いずれも長さが最小長ではないことがわかる。(さもなければそれは実効パスであり、多角形は分割されているはずだ)

Suppose we inset the boundary of the polygon by a distance h , as shown in figure 5.24. If the original vertices of the polygon were A_1, A_2, \dots then we will label the inset vertices A_1', A_2', \dots as we did for the gusset quad construction. I will call the inset polygon a **reduced polygon** of the original polygon.

図 5.24 のように、多角形の境界から距離 h だけ内側に多角形を挿入すると仮定する。マチ付き四辺形を構築したときと同じように、元となる多角形の頂点を A_1, A_2, \dots として、内部の頂点を A_1', A_2', \dots と名付ける。この挿入された多角形を、もとの多角形の縮小多角形と名付ける。

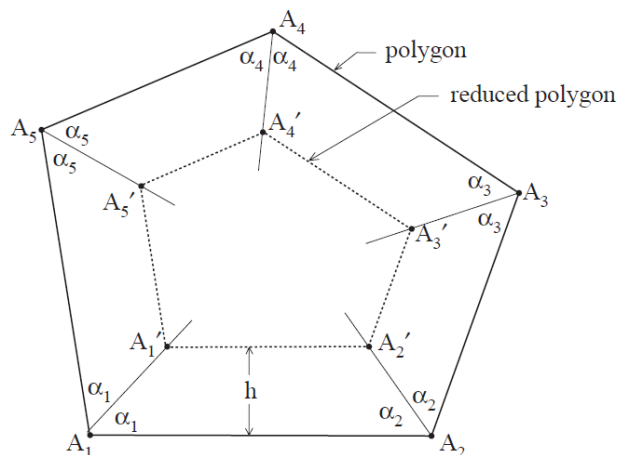


Figure 5.24. A reduced polygon is inset a distance h inside of an active polygon. The inset corners lie on the angle bisectors (dotted lines) emanating from each corner.

図 5.24 縮小多角形は、実効多角形から距離 h だけ内側に挿入される。挿入された頂点は、それぞれの頂点から伸びる二等分線（点線）上にある。

Note that the points A'_i lie on the bisectors emanating from the points A_i for any h . Consider first a reduced polygon that is inset by an infinitesimally small amount. In the folded base, the sides of the reduced polygon all lie in a common plane, just as the sides of the original active polygon all lie in a common plane; however, the plane of the sides of the reduced polygon is offset from the plane of the sides of the active polygon by a perpendicular distance h . As we increase h , we shrink the size of the reduced polygon. Is there a limit to the shrinkage? Yes, there is, and this limit is the key to the universal molecule. Recall that for any polygon that satisfies the tree theorem, the path between any two vertices satisfies a path length constraint

点群 A_i は、 h の値にかかわらず、点群 A_i から内側に引いた角の二等分線上にあることに注意しよう。まず、無限小の量だけ内挿された縮小多角形について考える。折りたたまれた基本形の中では、元となる実効多角形の辺が同一の平面に位置するとまったく同様に、縮小多角形の辺はすべて同一の平面上に位置する。しかし、縮小多角形の辺が位置する平面は実効多角形の辺が位置する平面から垂直に距離 h だけずらされている。 h の値を増加させるに従い、縮小多角形の大きさは減少していく。縮小の限界はあるのだろうか？答えは Yes である。そしてその限界値が、普遍的分子を解明する鍵となる。ツリー定理を満たす任意の多角形について、任意の二つの頂点間のパスは、パス長における次の制約を満たすことを思い出してほしい。

$$|A_i - A_j| \geq l_{ij} \quad (1)$$

where l_{ij} is the path length between nodes i and j measured along the tree graph. There is an analogous condition for reduced polygons; any two vertices of a reduced polygon must satisfy the condition

ここで、 l_{ij} は木グラフにそって節点 i と j の間の長さを計測したパス長である。縮小多角形にも似た条件がある。すなわち、縮小多角形における任意の二頂点は次のような条件を満たす。

$$|A'_i - A'_j| \geq l'_{ij} \quad (2)$$

where l'_{ij} is a **reduced path length** given by

l'_{ij} は次の式で得られる **縮小パス長** である

$$l'_{ij} = l_{ij} - h(\cot \alpha_i + \cot \alpha_j) \quad (3)$$

and α_i is the angle between the bisector of corner i and the adjacent side. I call equation (2) the **reduced path constraint** for a reduced polygon of inset distance h . Any path for which the reduced path constraint becomes an equality is, in analogy with active paths between nodes, called an **active reduced path**.

α_i は角 i の二等分線と隣接辺の間の角度である。方程式 (2) を、内挿距離が h の縮小多角形における**縮小パスの制約**と呼ぶことにする。縮小パスの制約で等号が成り立つ任意のパスを、節点間の実効パスにたとえて、**縮小実効パス**と呼ぶことにする。

So for any distance h , we have a unique reduced polygon and a set of reduced path constraints, each of which corresponds to one of the original path constraints. We have already assumed that all of the original path constraints are met; thus, we know that all of the reduced path constraints are met for the $h=0$ case (no inset distance). It can also be shown that there is always some positive nonzero value of h for which the reduced path constraints hold. On the other hand, as we increase the inset distance, there comes a point beyond which one or more of the reduced path constraints is violated. Suppose we increase h to the largest possible value for which every reduced path constraint remains true. At the maximum value of h , one or both of the following conditions will hold:

したがって、任意の距離 h について、縮小多角形が一意に得られ、縮小パスの制約の集合が一つ得られる。縮小パスの制約のそれぞれは元となるパスの制約のいずれかに対応する。元となるパスの制約がすべて成り立つことをすでに仮定しているため、すべての縮小パスの制約は $h=0$ (すなわち内挿なし) の場合にすべて成り立つことがわかる。また、 h の値として、縮小パスの制約を満たす、ゼロでない正のなんらかの値が常に存在することを示すことができる。その一方、内挿の距離を増やしてゆくと、ある点を越えたところから、一つまたは複数の縮小パスの制約が満たされなくなる。 h について、すべての縮小パスの制約が満たされる範囲で、とりうる最大値をもつように増加させると仮定する。この h の最大値においては、次の条件の片方または両方が満たされる。

- (1) For two adjacent corners, the reduced path length has fallen to zero and the two inset corners are degenerate; or
二つの隣接する頂点に関して、縮小パス長が 0 となり、内挿された二つの頂点が縮重する
- (2) For two nonadjacent corners, a path between inset corners has become an active reduced path.
二つの隣接していない頂点に関して、内挿された二つの頂点間のパスが縮小実効パスとなる

These two situations are illustrated in figure 5.25.
これら二つの状態は図 5.25 に示す

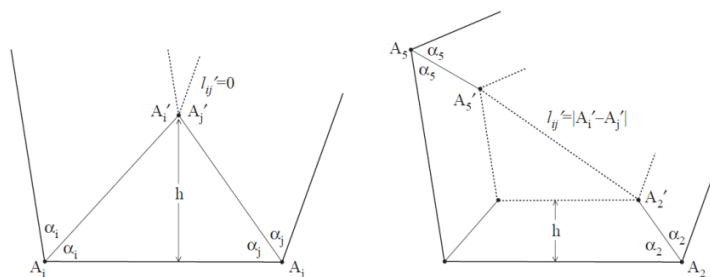


Figure 5.25. (Left) Two corners are inset to the same point, which is the intersection of the angle bisectors. (Right) Two nonadjacent corners inset to the point where the reduced path between the inset corners becomes active.

図 5.25 (左) 二つの頂点が同じ点に内挿される。この点は角の二等分線の交点である。(右) 二つの隣接していない頂点が、内挿された頂点間の縮小パスが実効パスとなるまで内挿される。

As I said, one or the other or both of these situations must apply; it is possible that path corresponding to both adjacent and nonadjacent corners have become active simultaneously or for multiple reduced paths to become active for the same value of h (this happens surprisingly often). In either case, the reduced polygon

can be simplified, thus reducing the complexity of the problem.

すでに述べたように、これら二つの状況がどちらか片方または両方満たされなければならない。同じ h の値で、隣接する二頂点が縮重すると同時に隣接していない二頂点に対応するパスが実効パスとなったり、あるいは複数の縮小パスが同時に実効パスとなったりするということが現実にもあり得る。(このようなことは驚いたことに、頻繁に発生する)。どちらの場合でも、縮小多角形を単純化することができるため、問題の複雑性を低下させることができる。

In a reduced polygon, if two or more adjacent corners have coalesced into a single point, then the reduced polygon has fewer sides (and paths) than the original active polygon. And if a path between nonadjacent corners has become active, then the reduced polygon can be split into separate polygons along the active reduced paths, each with fewer sides than the original polygon had (just as in the polygon network, an active path across an active polygon splits it into two smaller polygons). (In the gusset quad, for example, the reduced quad is inset until one of its diagonals becomes an active path; the reduced quad is then split along the diagonal into two triangles.) In either situation, you are left with one or more polygons that have fewer sides than the original. The process of inseting and subdivision is then applied to each of the interior polygons anew, and the process repeated as necessary.

縮小多角形において、もし二つ以上の隣接する角が合体して一つになっていたら、その縮小多角形は元の実効多角形よりも少ない数の辺(とパス)を持つ。そしてもし隣接していない角の間のパスが実効パスになった場合、その縮小多角形をその縮小実効パスに沿って別々の多角形に分けることができ、それぞれの多角形は元の多角形より少ない数の辺をもつ(多角形群において、実効多角形を横切る実効パスによってそれを二つの小多角形に分割したときと同様である)。(例えば四角マチの場合、縮小四辺形は対角線のいずれかが実効パスになるまで内挿されており、そのとき縮小四辺形は対角線で分割されて二つの三角形になる。)どちらの場合も元の多角形よりも辺の数が少ない多角形が一つ以上残される。それぞれの新たな内部多角形に対してさらに挿入や分割を適用し、必要に応じてその工程を繰り返す。

If a polygon (active or reduced) has three sides, then there are no nonadjacent reduced paths. The three bisectors intersect at a point, and the polygon's reduced polygon evaporates to a point, leaving a rabbit ear molecule behind composed of the bisectors.

もし(実効または縮小)多角形が三角形であるなら、隣接しない頂点に対応する縮小パスは存在しない。三つの二等分線は一点で交差し、その多角形の縮小多角形は消失して点となり、二等分線によって構成されるつまみ折り分子が残る。

Four-sided polygons can have the four corners inset to a single point or to a line, in which case no further inseting is required, or to one or two triangles, which are then inset to a point. Higher-order polygons are subdivided into lower-order ones in direct analogy.

四辺形は、四つの頂点が一つの点または一本の直線に内挿されるかもしれない。その場合、それ以上の内挿は必要ない。あるいは、四つの頂点が一つまたは二つの三角形に内挿されることもあるが、その三角形は1つの点に内挿される。より高次の多角形の場合は、同様の方法で低次のものに分割する。

Since each stage of the process absolutely reduces the number of sides of the reduced polygons created (although possibly at the expense of creating more of them), the process must necessarily terminate. Since each polygon (a) can fold flat, and (b) satisfies the tree theorem, then the entire collection of nested polygons must also satisfy the tree condition. Consequently, *any* active polygon that satisfies the tree theorem — no matter how many sides — can be filled with a crease pattern using the procedure outlined above and collapsed into a base on the resulting creases.

それぞれの工程で、生成される縮小多角形の辺の数は必ず減らされるため(その代わり多角形の数は増えるかもしれないが)、この工程は必ず終わるに違いない。それぞれの多角形が平らに折り畳め(a)、ツリー定理を満たす(b)ので、入れ子にされた多角形の集合全体もツリーの条件を必ず満たす。したがって、ツリー定理を満たす「任意の」実効多角形は、何角形であっても一上で解説された手順を用いて展開図で埋めることができ、その結果として得られた折り線で基本形へと折ることが出来る。

So what are the creases of the universal molecule? Each polygon is divided into two parts: the **core** is the reduced polygon (which may be crossed by active reduced paths); the border around the core is the **ring**. The angle bisectors that cross the ring are mountain folds. Internal nodes along active paths propagate inward across the ring forming tri-state folds. Active reduced paths that cross the core are valley folds. The boundary of reduced polygons can also be tri-state folds, as one may or may not fold layers along them. The same assignment of crease applies to each level of the recursive universal molecule construction.

それでは、普遍的分子の折れ線はどのようなになるだろうか？一つの多角形は二つの部分に分けられている。縮小多角形（縮小実効パスで区切られていることもある）を**核**とし、核のまわりの境界領域を**輪**とする。輪を横切っている角の二等分線は山折りである。実行パス上の内節点は、輪を横切って内挿され、三状態の折り線を形成する。核を横切る縮小実効パスは谷折りである。縮小多角形の境界も、折られる場合もあれば折られない場合もあるので、三状態の折り線である。折り線に関する同様の割り当てが、それぞれのレベルの普遍的分子の構造に再帰的に適用される。

A remarkable feature of the universal molecule is that many of the molecular crease patterns that have been previously enumerated are just special cases of it, including the rabbit ear molecule, the gusset quad, and both Maekawa and Meguro quads. Figures 5.26 and 5.27 illustrates these special cases.

普遍的分子に関して特筆すべき点としてあげられるのが、つまみ折り分子、マチ付き四辺形分子、前川四角形と目黒四角形の両方の分子を含む、上で挙げた分子の展開図の多くが、普遍的分子の特殊例に過ぎないという点である。図 5.26 と 5.27 にそれらの特殊例を示す。

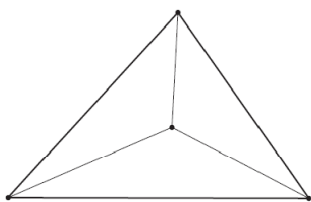


Figure 5.26. In a triangle, all three corners are inset to the same point, which is the intersection of the angle bisectors. This gives the rabbit ear molecule.

図 5.26 三角形においては、三つの頂点は同一の場所に内挿され、その場所は角の二等分線の交点である。これはつまみ折り分子になる。

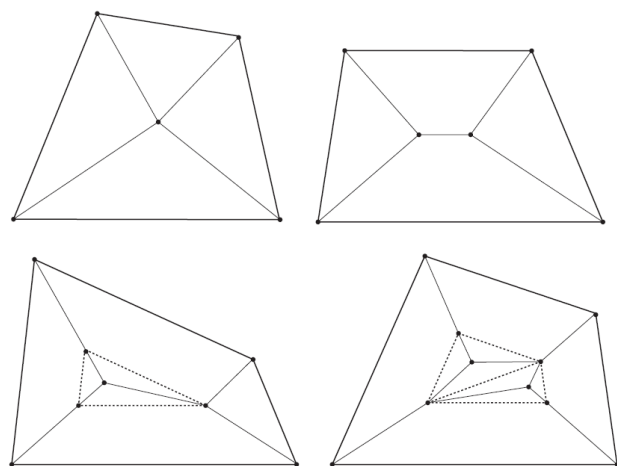


Figure 5.27. The four possible universal molecules for a quad. If all four corners are inset to the same point, the result is the Husimi molecule (top left). If adjacent pairs of corners are inset to two points, the Maekawa molecule is obtained (top right). If the inset polygon is a triangle, it is filled in with a rabbit ear

molecule, which also results in a Maekawa molecule (bottom left). Finally, if the inset polygon is a quad crossed by an active reduced path, the result is the gussetquad.

図 5.27 四角形においてありうる 4 種類の普遍的分子。もし 4 つの頂点全てが同じ場所に内挿された場合、その結果は伏見分子（左上）となる。もし隣接する頂点の組が二点に内挿された場合、前川分子が現れる（右上）。もし内挿された多角形が三角形なら、そこにはつまみ折り分子が挿入され、同様に前川分子となる（左下）。最後に、もし内挿された多角形が、アクティブな縮小パスが横切る四角形であった場合、結果はマチ付き四辺形となる。

These examples are just the tip of the iceberg; beyond four sides, the possibilities rapidly explode. But this explosion doesn't matter; there is a unique universal molecule for *every possible* active polygon that satisfies the tree theorem.

これらの例は氷山の一角にすぎない。辺の数が四本を超えると、可能な分子の展開図が急速に増加する。しかしこの増加は問題にならない。ツリー定理を満たすような「存在しうるすべての」実効多角形のそれぞれについて、普遍的分子はただ一つだけ存在する。

An alternative design approach that blends aspects of the circle method and tree methods has been described by Kawahata² and Maekawa³. It has been called this the “string-of-beads” approach to design. As in the tree method, you begin with a tree graph of the model to be folded. Each line of the graph is doubled and the graph is expanded to fill a square, with the nodes of the graph spaced around the edges of the square like beads on a string. The process is illustrated for a six-flap based in figure 5.28.

円領域分子法とツリーメソッドの手法を混合した、別の手法が川畑[2]と前川[3]によって示されている。これを「数珠つなぎ」設計手法と呼ぶことがある。ここでも、ツリーメソッドのように、折ろうとする作品の木構造のグラフから開始する。グラフのそれぞれの線を 2 重にし、グラフを広げて正方形を満たす。グラフの節点はひもで結ばれた数珠のように正方形の辺上に並ぶ。図 5.28 にその過程を六つのカドをもつ基本形に適用した場合を図示する。

[2] Fumiaki Kawahata, *Fantasy Origami*, pub. by Gallery Origami House, Tokyo, Japan, 1995 [in Japanese].
川畑文昭、空想おりがみ、ギャラリーおりがみはうす、東京、日本、1995

[3] Jun Maekawa, *Oru* magazine, also in Proceedings of the Second International Conference on Origami Science and Technology, 1994 [in Japanese]
前川淳、季刊『をる』および Proceedings of the Second International Conference on Origami Science and Technology, 1994

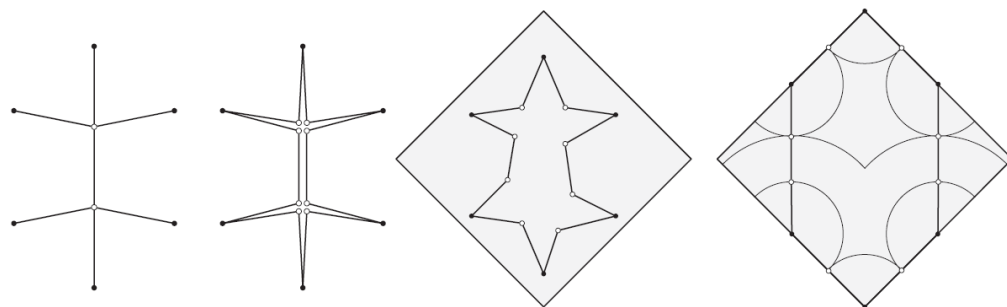


Figure 5.28. The string-of-beads design method. The tree graph is turned into a closed polygon, which is then “inflated” inside of a square with straight lines between the terminal nodes. The result is a large polygon inside the square that is collapsed into the base.

図 5.28 数珠つなぎ設計手法。木構造を閉じた多角形に変換し、端節点間が直線になるように正方形の中で「膨らませる」。結果として基本形に折り畳める大きな多角形が正方形内に現れる。

In the string-of-beads method, the tree graph is converted into a large polygon in which each corner is one of the terminal nodes of the tree and each side is as long as the path between adjacent terminal nodes. It is clear that this distribution of terminal nodes is just a special case of the tree method in which we have constrained all of the nodes to lie on the edge of the square; it avoids having middle points, but at the expense of possibly reduced efficiency.

数珠つなぎ設計手法においては、木グラフは一つの大きな多角形へ変形させられる。それぞれの頂点が木の端節点のいずれかであり、それぞれの辺の長さは隣接する端節点間のパスの長さに等しい。端節点群のこの配置は、正方形の辺上にすべての端節点をのせるという制約を設けた場合の、ツリーメソッドの特殊例にすぎないことは明らかである。この場合、内部にカドを配置することを避けることができるが、その代わり紙の効率が低減する可能性が生じる。

The string-of-beads approach produces a large polygon that must be collapsed into the base, and the techniques described by Maekawa involve placing tangent circles in the contours shown in the last step of figure 5.28 (which is analogous to our use of additional circles to break down active polygons into smaller polygons in the tree method; Kawahata's algorithm projects hyperbolas in from the edges to locate reference points for molecular patterns.) However, one can also apply the universal molecule directly to the string-of-beads polygon, achieving another efficient crease pattern that collapses it into a base.

数珠つなぎ手法によって生成される多角形は基本形へ折り畳むことが出来るはずであるが、前川によって示された方法では、図 5.28 の最後のステップに図示されるような互いに輪郭で接する円の配置を使用する（これはツリーメソッドにおいて実効多角形をより小さな多角形へと分割する際に用いた追加の円の使用によく似ている。川畑のアルゴリズムでは、辺から内部へ双曲線を伸ばし、分子の展開図の基準点を見つける）。しかしながら、普遍的分子も同様に数珠つなぎの多角形に直接適用することができ、基本形へと折り畳める別の効率的な展開図が得られる。

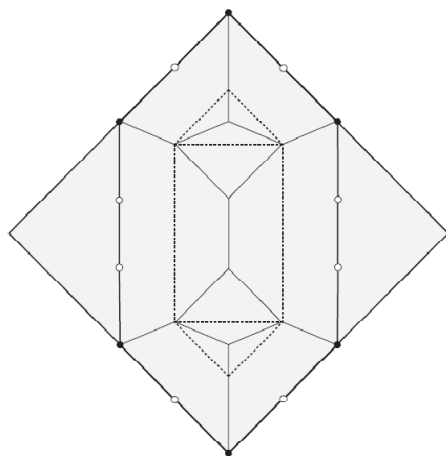


Figure 5.29. Construction of the universal molecule for the polygon shown in figure 28.

図 5.29 図 28 に示された多角形の普遍的分子の作図

Figure 5.29 shows the universal molecule. The initial hexagon is inset to the point that the two horizontal reduced paths become active, and the hexagon is split into two triangles and rectangle. The triangles are filled with rabbit ear creases; the rectangle is further inset, forming a Maekawa molecule.

図 5.29 がその普遍的分子を示している。最初の六角形が、二本の水平な縮小パスが実効パスとなるまで内挿され、縮小された六角形が二つの三角形と長方形に分割される。三角形はつまみ折りの展開図で満たされ、長

方形はさらに内挿されて前川分子を形成する。

Figure 5.30 compares the crease pattern obtained from this polygon by adding an additional node to the tree pattern (i.e., adding a middle flap) and that obtained with the universal molecule. (Tristate creases are shown as dotted lines.)

図 5.30 は、木構造に節点を追加することによって（すなわち、内部のカドを加えることで）得られた多角形の展開図と、普遍的分子によって得られた展開図を比べたものである（三状態の折り目は点線で表現されている）。

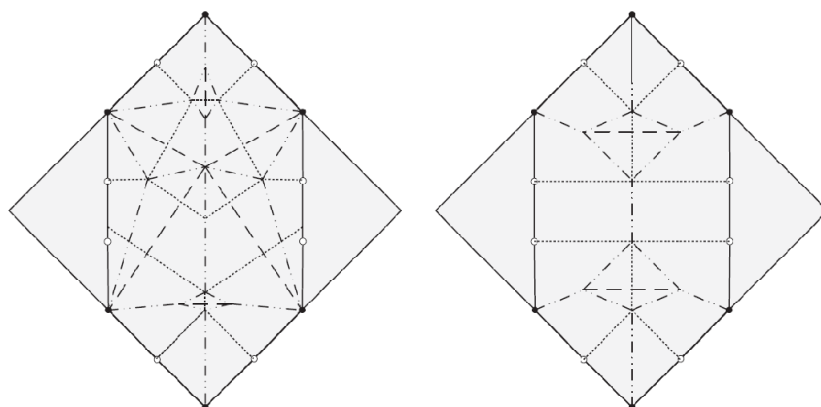


Figure 5.30. (Left) Crease pattern obtained by adding an additional node to the hexagonal active polygon. (Right) Crease pattern obtained by using the universal molecule.

図 5.30（左）六角形の実効多角形に節点を追加することで得られた展開図（右）普遍的分子を用いることで得られた展開図

A nice feature of the universal molecule is that it is very frugal with creases. A tree filled in with universal molecules tends to have relatively few creases and large, wide flaps (which can, of course, be subsequently narrowed arbitrarily as desired). In fact, I conjecture the following: for any active polygon, the universal molecule is the crease pattern with the shortest total length of creases that collapses that polygon to the uniaxial base. Few creases translates into relatively few layers in the base (at least, until you start sinking edges to narrow them). And because you don't have to arbitrarily add circles (and hence points) to a crease pattern to knock polygons down to quads and triangles (as you do using the classical tree method algorithm), bases made with the universal molecule tend to have less bunching of paper and fewer layers near joints of the base, resulting in cleaner and (sometimes) easier-to-fold models.

普遍的分子の利点は、折り目の数を節約できるという点である。木に対して普遍的分子を用いると、折り目の数が比較的少なく、カドは広く、太くなる傾向がある（もちろん、必要であれば好みに応じて細くすることが可能である）。実際、私は次のように推測している：任意の実効多角形において、普遍的分子は多角形を単軸基本形に折り畳むための折り目の長さの和が最も小さくなる展開図である。折り目が少ないということは、基本形における紙の重なりが相対的に少ないということである（少なくとも、縁を沈め折りして細くし始めるまでは）。そして多角形を四角や三角に細分化するために（古いツリーメソッドのアルゴリズムを使うときのように）任意の円（つまりカド）を追加しなくても良いため、普遍的分子で出来た基本形は紙の固まりがより少なくなり、また基本形の結合部付近の重なりが少なくなり、結果的にわかりやすく（時には）折りやすい作品ができる傾向がある。

As another example of the utility of the universal molecule, an article by Maekawa in *Oru* magazine illustrated the design of a tree structure using the string-of-beads/contour algorithm. In this algorithm, all terminal nodes are arranged around the outside of the square to form one large active polygon. Then fixed-size circles are added to the interior to break up the active polygon into quads and triangles. Figure 5.31 shows the tree and the circle pattern derived thereby.

上で述べたように、追加する内部円の数減らし、新しい円を加えるたびに最大の大きさまで膨らましてから次の円を追加することで、同じ端節点の配置に対して得られる展開図を単純化することが出来る。実際、たった三つの節点を追加するだけで、すべての実効多角形が三角形か四角形であるようにすることができる。図 5.33 に示されるように木と展開図を修正した結果は折るのがとても簡単で、つまみ折りとマチ付き四辺形の分子のみが利用されている。

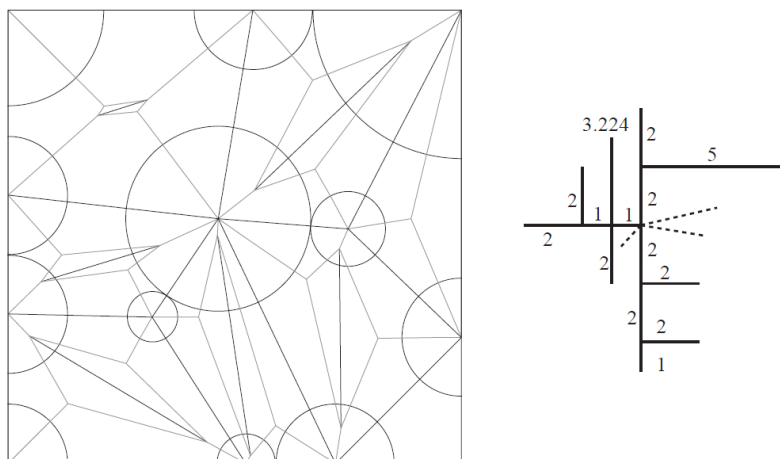


Figure 5.33. Tree and crease pattern for the *Oru* tree with three additional interior circles and filled with triangle and gusset quads.

図 5.33 『をる』の木に三つの内部円が追加され、三角形とマチ付き四辺形で構成された木と展開図

By applying the universal molecule to the original polygon, we can get a still simpler crease pattern, which is shown in figure 5.34. This pattern has the additional perk that it is yet easier to fold up in the base (I encourage you to try). The flaps are, as you might expect, much wider than they are long. However, by sinking the top of the base in and each, all of the flaps can be thinned to arbitrarily large aspect ratio while maintaining their relative lengths.

元となる多角形に普遍的分子を適用することで、図 5.34 に示すような、さらに簡単な展開図が得られる。このパターンには基本形がさらに簡単に折り上げられるという特典がある（実際に折ってみることを奨励する）。予想した通り、カドは長さ比べてずっと幅広になっている。しかし、基本形の頭頂部をそれぞれ沈め折りしてやることで、全てのカドを相対的な長さを維持したまま任意の縦横比に細くすることが出来る。

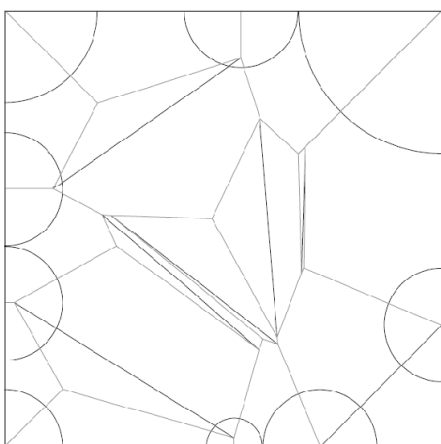


Figure 5.34. Universal molecule version of the *Oru* tree

図 5.34 『をる』の木の普遍的分子バージョン

All of the crease patterns in figures 5.31–5.34 share the property that all points lie on the edge of the square. If we relax this constraint and allow middle points, then we can achieve a slightly larger and even simpler pattern (with a scale of 0.067 as compared to 0.065 for the previous pattern), shown in figure 5.35.

図 5.31–5.34 の全ての展開図は、正方形の輪郭の上にすべての点があるという特徴を共有している。もしこの制約を緩和し、内部の点を許容した場合、図 5.35 に示すような、わずかに大きくさらに簡単な展開図（前のパターンの 0.065 倍と比較して、0.067 倍）を得ることが出来る。

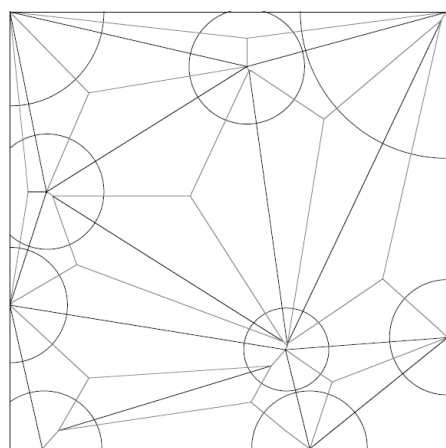


Figure 5.35. *Oru* tree crease pattern with middle points.

図 5.35 内部に点のある『をる』による木の展開図

Without the universal molecule one must arbitrarily add circles to the network of active paths until every polygon is of order 4 or smaller. Since there is some choice about where circles are added, the tree method solution for a base is not necessarily unique. The universal molecule, however, is entirely determined. By applying the recursive universal molecule construction, any network of active paths can be filled in with creases and collapsed into a uniaxial base. Thus, the projection from tree graph to full crease pattern may be accomplished using a single optimization.

普遍的分子がなければ、それぞれの多角形の頂点が 4 以下になるまで実効パスに任意の円を追加しなければならない。円を加える場所についていくつかの選択肢があるので、ある基本形についてのツリーメソッドによる解法は一意でない。しかし、普遍的分子は完全に確定している。普遍的分子の構築を再帰的に適用することで、どのような実効パス網であっても折り目で埋めることができ、単軸基本形に折り畳むことが出来る。したがって、木グラフから完全な展開図へと進めることは一つの最適化をおこなうだけで達成することが可能だ。

There is another, subtler advantage to the universal molecule: it tends to shift paper away from the plane of projection of the base, which simplifies collapsing the base and reduces the buildup of layers where flaps join together. Although the flaps tend to be wider than those produced with other algorithms, one can easily make points narrower by repeatedly sinking the corners of collapsed polygons to narrow the flaps. This, of course, builds up layers of paper in the base, which is to some degree unavoidable. The most equitable distribution of these extra layers comes when the points are sunk along lines parallel to the plane of projection. Since the edges of the inset polygons are parallel to the plane of projection in the folded base, these lines naturally form lines along which to sink.

他にも、普遍的分子に関する次のような小さな利点がある：普遍的分子には、紙を基本形の射影平面からはずれたところに配置するという傾向があるので、基本形の折り畳みが簡略化され、カドが互いに接続する場所の

折り重なりが少なくなる。他のアルゴリズムで生成されたものと比べてカドはより幅広くなる傾向があるが、折り畳まれた多角形の頂点を沈め折りして細いカドにすることを繰り返すことで、そのカドをより細くすることが簡単にできる。もちろん、これによって基本形の紙の層が折り重ねられるが、それはある程度避けられないものである。これら余分な層の配置のうち最も均等なものは、カドが射影平面と平行な線にそって沈められたときである。内挿多角形の辺は折り畳まれた展開図の射影平面と平行であるため、これらの線は自然に沈め折り出来る線を形成する。

There are other polygon-filling algorithms beyond the universal molecule described above, and I will mention two of them here. Toshiyuki Meguro — who coined the term “molecule” (*bun-shi*, in Japanese) has developed a technique that makes use of overlapping circles, which in effect, allows one to add new branches to the tree by adding nodes to the middle of existing edges. This technique allows each new circle to touch four, rather than three, existing circles, and cuts down on the number of circles that need to be added. I have extended and generalized Meguro’s concept to apply to arbitrary polygons by introducing the concept of a “stub.” Also Fumiaki Kawahata has developed a technique for filling in polygons that involves projecting hyperbolas, rather than straight lines, inward from the edges of an active polygon, which results in bases whose points can be narrower and more regular (if desired). No doubt there are yet other algorithms lurking out there in the mathematical wilds.

上記で説明した普遍的分子以外にも、多角形を満たすための他のアルゴリズムが存在するので、ここでそれらの二つについて述べる。「分子」という言葉を初めて使った目黒俊幸は、互いに重なる円を使い、既にある辺の中間に節点を追加することで木に新しい枝を追加することを可能にする技法を開発した。この技法では、新たな円を、三つではなく四つの既存の円に接するように追加することができるので、追加する必要のある円の数が削減される。私は、「スタブ」という概念を導入することで、目黒の考えを拡張し任意の多角形に適用するように一般化した。また、川畑文昭は実効多角形の辺から内側に向かって、直線ではなく双曲線を引くことでその多角形を埋める手法を開発し、その結果基本形のカドは（望むなら）より細くより規則的になる。他にも、未開拓の数学の世界に隠れたアルゴリズムが未だに存在しているのは間違いない。

Although the mathematical algorithms for origami design are rigorously defined, the actual location of the nodes and creases can be computationally intensive. Computationally intensive problems are best handled by computer and indeed, the procedures described above can be cast in the mathematical and logical terms that lend themselves to computer modeling. The computer program *TreeMaker* implements these algorithms. Using *TreeMaker*, I’ve solved for bases for a number of subjects whose solutions have eluded me over the years — deer with varying sizes and types of antlers, 32-legged centipedes, flying insects, and more. Using a computer program accelerates the development of a model by orders of magnitude; from the tree to the full crease pattern takes less than five minutes (although folding the crease pattern into a base may take two to three hours after that!) Not only does *TreeMaker* come up with the base initially, but it lets one incrementally iterate the design of the model, shifting paper from one part of the model to another to lengthen some points and shorten others, all the while keeping the entire model maximally efficient.

このような折り紙創作のための数学的アルゴリズムは厳密に定義されているが、節点と折り線の実際の位置を決めるには、複雑な数値計算が必要となるだろう。複雑な数値計算が必要とされる問題に対してはコンピュータを用いることが一番で、実際、上記の手順を数学と論理で記述し、コンピュータモデルに変換することが可能である。コンピュータプログラム「TreeMaker」にはそのようなアルゴリズムが実装されている。TreeMakerを用いることで、数々の対象-さまざまな形と大きさの角を持った鹿、32脚のムカデ、飛ぶ虫など-の基本形を求めるという、数年以上解決できなかった問題を解決することができた。コンピュータプログラムを用いることで、作品作成の速度が桁違いに加速した。木から完全な展開図を作成するまでが5分以下しかかからない(そのあと展開図を基本形へ折り畳むのに二時間から三時間がかかるかもしれないが!)。TreeMakerは最初の基本形を生み出すためだけでなく、作品のある場所から紙を別の場所にずらして、あるカドの長さを伸ばすと同時に別のカドを縮めるなど、作品の設計を繰り返し修正するためにも使え、その間でもずっと作品全体は最も効率の良い状態に保たれる。

One need not program a computer, of course, to use the techniques I’ve described to design origami — origami designers have used similar techniques for years. But I foresee a dramatic shift in the art as these techniques — what one might call “algorithmic” origami design — become more widespread. For years, technical folders concentrated on getting the right number and lengths of points to the near-exclusion of other folding considerations such as line, form, and character. With algorithmic origami design, point

count comes automatically and is no longer the overwhelming consideration in technical origami design. In the past, origami art and origami science have often been at odds, but with algorithmic origami design, the technical designer, freed from the need to expend his or her energies on the number of appendages, can focus on the art of folding, secure in the knowledge that the science will take care of itself.

もちろんコンピュータをプログラムせずとも、すでに説明した折り紙の創作手法を使うことができる。折り紙作家はすでに同様の手法を何年も使ってきた。しかし、「アルゴリズム」折り紙創作とも呼びうるこれらの技法がより広まるにつれて、この芸術分野において大きな改革が起きると私は予想している。長年、折り紙設計を用いる作家たちは正しい数と長さのカドを得ることに集中し、線や形、特徴などといった他の折り紙の問題をほとんど除外していた。アルゴリズム折り紙創作を行うことで、カドの数は自動的にもとまり、それはもう折り紙創作技法における困難な問題ではなくなった。過去において、芸術折り紙と科学折り紙はしばしば不仲であったが、アルゴリズム折り紙創作によって、設計を用いる作家は、カドの数で労力を使い果たす必要性から解放され、科学が面倒を見てくれるということを知ることによって、安心して折り紙の芸術性に焦点を合わせることができるのである。

Acknowledgment

本文書の翻訳にあたり、羽鳥公士郎様には稚拙な訳文を訂正していただくなど多大なる協力をしていただきました。ここに心より感謝の意を表します。どうもありがとうございました。

2008.7.27 古田陽介