NASA

Earth Science Data Systems
Software Reuse Working Group
Technical Report SRWG-TR-2010-001

Mathematical Theory of Reusability

Michael Leyton

DIMACS Center for Discrete Mathematics & Theoretical Computer Science
Busch Campus, Rutgers University, New Brunswick, NJ 08854, USA.
mleyton@dimacs.rutgers.edu

ABSTRACT

The book *A Generative Theory of Shape* (Michael Leyton, Springer-Verlag, 2001) develops New Foundations to Geometry specifically designed to give a single mathematical language for the entire range of software and data objects, in the product lifecycles and data lifecycles, of large-scale engineering and scientific systems. This mathematical language is based on what this theory regards as the two fundamental principles of intelligence: Maximization of Transfer and Maximization of Recoverability of the generative operations that produced the object. The New Foundations to Geometry give a mathematical theory of transfer and a mathematical theory of recoverability. Furthermore, the foundations combine these two mathematical theories, and this leads to a Mathematical Theory of Intelligence. This Mathematical Theory of Intelligence structures objects in such a way that they become maximally reusable, interoperable, and archival.

The reason is this: The theory claims that reusability of an object is maximized if the object itself is defined as having been produced by maximizing reuse of the operations that were used to produce it. This is because reuse *within* the object achieves most of the reuse that is needed when the entire object has to be reused. Therefore the object must be represented generatively; and the generative operations used to represent it must be maximally reused in that representation. In the New Foundations to Geometry, maximization of reuse of the generative operations is called Maximization of Transfer. Furthermore, to ensure the maximization of reuse of the generative operations, the operations must be maximally *recoverable*. Thus, the maximization of reuse is dependent on the maximization of recoverability. Therefore, according to the New Foundations to Geometry, in order to ensure maximization of reusability of an object, it must be given a representation that accords with the two basic principles: Maximization of Transfer and Maximization of Recoverability.

Mathematically, the New Foundations model *transfer* by a group-theoretic structure called a wreath product. Furthermore, according to the New Foundations, generative operations are recoverable only if they are symmetry-breaking. The New Foundations give an entirely new theory of symmetry-breaking, in which symmetry-breaking is modeled by the *transfer* of the past symmetry onto the present broken symmetry; i.e., the *reuse* of the past symmetry. As a result of this, the combination of transfer and recoverability leads to a powerful mathematical structure, invented in the New Foundations, called: *symmetry-breaking wreath products*.

According to the New Foundations, the *conventional* theory of symmetry-breaking, in physics and chemistry, is bad for the needs of Science Data Systems because it defines the group of the current symmetry-broken state as a reduced version of the past symmetry group, and therefore looses the past state of the object as well as the object's history. In contrast, the Theory of Symmetry-Breaking invented by the New Foundations to Geometry is good for the needs of Science Data Systems because it represents the data in terms of a recovered generative process that transfers the past symmetry onto the current state, and thus gains the past state of the object as well as the object's history. Therefore, because this defines the object in terms of the causal processes that produced it, the data is converted into a form that is useful for scientific research, as required by Science Data Systems. Furthermore, in this new Theory of Symmetry-Breaking, the

raw science data is given a structure that is a *reuse* of past structure. That is, since the past symmetry is not lost, but is transferred onto the present asymmetric state, this means that the present builds on the past, rather than throwing the past away, and the past state is *reused*.

Because the New Foundations to Geometry are based on the Maximization of Recoverability principle, the data representations are given a format which optimizes their capacity for *archiving*. According to the New Foundations to Geometry, the Standard Foundations to Geometry are bad for archiving because the Standard Foundations are based on the *invariants* program, which defines geometric objects as invariants, i.e., the properties that remain unchanged by actions, and therefore are memoryless with respect to those actions. In opposition to this, the New Foundations to Geometry define geometric objects as memory stores. Furthermore, the New Foundations are the only theory that have given a mathematical theory of memory storage. According to the New Foundations, any memory store is structured as a symmetry-breaking wreath product. As a result of this, the New Foundations bring *reuse* into the structure of memory stores.

A crucial role of the New Foundations is that they give a comprehensive theory of *raw data representation* such that the data set is maximally reusable in the *data lifecycle*. The book *A Generative Theory of Shape* elaborates this theory in detail, and comprehensively explains the use of this theory to represent *scientific data* and *manufacturing data*. The present paper gives a summary of some parts of this theory as well as illustrations in terms of computer-aided design, kinematics, general relativity, quantum mechanics, natural morphology, geodesic polar coordinates, software engineering, etc.

As stated in the book *A Generative Theory of Shape*, the fundamental purpose of the New Foundations to Geometry is to handle *complexity*. This is achieved by a class of mathematical groups, invented in the New Foundations, called *unfolding groups*. Major classes of unfolding groups are structured by starting with a configuration in which n primitives are maximally aligned. This configuration is called the *alignment kernel*. The unfolding causes successive and selective misalignment of the primitives. Because this works by transfer, the unfolding action maps the alignment kernel onto misaligned versions of itself. Thus, in unfolding groups, the misaligned versions are mathematically described as the *reuse* of the original aligned state.

As an example, according to the New Foundations to Geometry, shape bifurcation, which is a crucial aspect of morphology (e.g., in geology, meteorology, biology, etc.), is mathematically structured by unfolding groups, which the New Foundations invented to describe complexity in terms of *reuse*.

Using the above concepts, this paper shows how the New Foundations to Geometry give *New Foundations to Object-Oriented Programming*, including inheritance, object-creation, class structure, class consistency, command structure, software text, etc.

1 New Foundations to Geometry

My book *A Generative Theory of Shape* **(Springer-Verlag, 2001) develops New Foundations to Geometry specifically designed to give a single mathematical language for the entire range of software and data objects, in the product lifecycles and data lifecycles, of large-scale engineering and scientific systems.**

This mathematical language is based on what this theory regards as the two fundamental principles of intelligence:

(1) Maximization of Transfer. Any agent is regarded as displaying intelligence and insight when it is able to *transfer* actions used in previous situations to new situations. In fact, the agent must maximize the transfer of parts of generative sequences onto other parts of generative sequences.

(2) Maximization of Recoverability. Any intelligent agent must be able to infer the causes of its own current state, in order to identify why it failed or succeeded, and thereby edit its behavior. Notice that this is part of a still larger problem, which the theory calls the problem of recoverability: Given the present state of an object, recover the sequence of operations which generated that current state.

My New Foundations to Geometry give a mathematical theory of transfer and a mathematical theory of recoverability.

Furthermore, the foundations combine these two mathematical theories, and this leads to a Mathematical Theory of Intelligence.

This Mathematical Theory of Intelligence structures objects in such a way that they become maximally reusable, interoperable, and archival.

To demonstrate the power of the New Foundations to Geometry, the book shows that this Mathematical Theory of Intelligence formalizes a large array of scientific and technical disciplines, including software engineering, robotics, computer-aided design, general relativity, quantum mechanics, mechanical engineering, the theory of differential equations, computer vision, the areas of human cognitive science, etc. Furthermore, in doing so, it discovers and exhibits fundamental correspondences between all these disciplines, thus allowing a single language to handle the multi-disciplinary nature of the software and data objects involved in product lifecycles and data lifecycles.

2 Mathematical Theory of Transfer

To describe this Mathematical Theory of Intelligence, we will first describe the Mathematical Theory of Transfer. This begins by giving a statement of the Principle of the Maximization of Transfer, which the reader will recall, is basic to the above *definition of intelligence*.

MAXIMIZATION OF TRANSFER. *In the generative sequence defining an object, make one part of the generative sequence a transfer of another part of the generative sequence, whenever possible. Therefore the maximization of transfer gives a* **maximization of reuse** *of the generative operations that define the object.*

Notice that this definition has the following consequence:

MODULARIZATION: *The maximization of transfer of the operations that generate the object has the effect of modularizing the object into a hierarchy of generative components that are* **maximally reused** *within the generation of the object.*

We will see that this particular modularization enables the object and its components to be maximally adaptable and editable for integration into other systems. That is, our theory has the following important claim:

INSIDE-TO-OUTSIDE MAXIMIZATION OF REUSE: *Representation of the object as generated by operations that were maximally reused* within *the generation of the object, makes the object itself maximally reusable* elsewhere. *This is because reuse* within *the object achieves most of the reuse that is needed when the entire object has to be reused.*

Now let us begin to understand the Mathematical Theory of Transfer. According to this theory, a situation of transfer involves two levels as illustrated in Fig 1: a *fiber group*, which is the group of actions to be transferred; and a *control group*, which is the group of actions that will transfer the fiber group. The transferred versions of the fiber group are shown as the vertical copies in Fig 1, and will be called the *fiber-group copies*. The control group acts from above, and transfers the fiber-group copies onto each other, as indicated by the arrow.

Notice that the structure described captures a property of the **reuse** structure in **product-line engineering**. Later, this paper will show how other aspects of product-line engineering are captured in this structure, in particular by the new classes of groups invented in the New Foundations to Geometry in order to handle complexity. These groups are called *unfolding groups*, and they have three main subclasses: *telescope groups*, *super-local unfoldings groups*, and *sub-local unfolding groups*. These model the different types of variability in product-line engineering, e.g., design inherited state machines, design parameterized state machines, etc. These groups also provide an advanced theory of **modularization**.

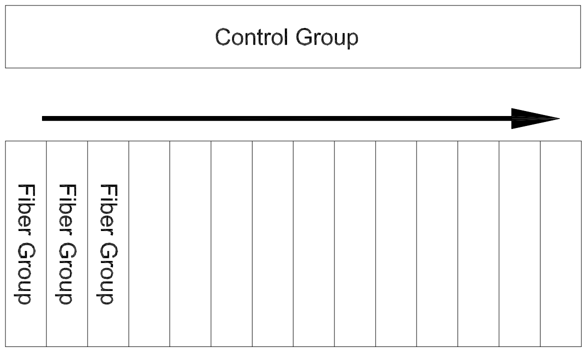

Figure 1: The control group transferring the fiber-group copies onto each other.

Now let us describe the Mathematical Theory of Transfer *in detail*. The basic claim is that a situation of transfer is built from two group actions. On the lower level, there is an action of a group $G(F)$, which we will call a **fiber group**, on a set F, which we will call a **fiber set**. On the upper level, there is an action of a group $G(C)$, which we will call a **control group**, on a set C, which we will call a **control set**.

Then, to model transfer, make the transferred versions of the fiber group, as follows: For each member c in the control set, make a copy $G(F)_c$ of the fiber group $G(F)$. These will be the transferred versions, and will be called the **fiber-group copies**, shown as the columns in Fig 1. Most crucially, the action of the control group $G(C)$ on the control set C can therefore be *imitated* by an action of the control group $G(C)$ on the collection of fiber-group copies. It is this imitating action that will be regarded as the transferring action that the control group has on the fiber-group copies, i.e., this will be regarded as sending the fiber-group copies onto each other. That is:

The fiber-group copies are the <u>reused versions</u> of the fiber group.

A fundamental property of the Mathematical Theory of Transfer is that it pulls all these components together into a single encompassing structure. According to the theory, the encompassing structure is best given by a group-theoretic construct called a *wreath product*, which is defined as follows: Intuitively, a wreath-product is a group that contains the entire structure shown in Fig 1. The structure of this total group is as follows: In Fig 1, the entire lower block shown is the direct product $\prod_{c \in C} G(F)_c$ of the fiber-group copies. We will call this the **fiber-group product**. The wreath product group is then built up by adding, to the fiber-group product, the control group $G(C)$ by what is called a *semi-direct product*, explained as follows:

In any semi-direct product, the upper group (here the control group) sends the lower group (here the fiber-group product) to itself by rearrangements that preserve the latter's group structure. Such rearrangements are called automorphisms. In a wreath product, this automorphic action is one in which the control group sends the fiber-group copies onto each other in a way that exactly imitates the action of the control group on the control set.

To state this rigorously: One constructs an **automorphism representation**

$$\tau : G(C) \longrightarrow Aut\{\prod_{c\in C} G(F)_c\}$$

such that, given an element g in the control group, its effect on the fiber-group product is defined thus:

$$\tau(g) : \prod_{c\in G(C)} G(F)_c \longrightarrow \prod_{c\in G(C)} G(F)_{g^{-1}c}.$$

From this automorphism representation, one can then construct the corresponding external semi-direct product:

$$\{\prod_{c\in C} G(F)_c\} \circledS_\tau G(C).$$

To understand this notation, notice that, to the left of the semi-direct product symbol Ⓢ is the fiber-group product, i.e., the entire bottom block we diagramed in Fig 1. To the right of the Ⓢ symbol is the control group $G(C)$, the upper block diagramed in Fig 1. Notice also that the subscript τ on the symbol Ⓢ is the automorphism representation which defines what I call the *transfer* effect of the control group on the fiber-group copies.

It is this semi-direct product that is the *wreath product* of the fiber group and the control group, written like this:

$$\begin{aligned}\textbf{Fiber Group Ⓦ Control Group} &= G(F) \text{ Ⓦ } G(C) \\ &= \{\prod_{c\in C} G(F)_c\}\circledS_\tau G(C). \qquad (1)\end{aligned}$$

Let us now understand how the New Foundations to Geometry model transfer within the wreath product. The claim is that transfer corresponds to what is algebraically called conjugation, $g\phi g^{-1}$ where g is a member of the control group and ϕ is a member of the fiber-group product. Most crucially, let us understand its effect on the fiber-group copies. Notice that each fiber-group copy has an embedded version within the wreath product. We can call this, the *embedded fiber-group copy*; and, often, for convenience we will simply call it, the *fiber-group copy*. Similarly, the control group has an embedded version within the wreath product. Again, we can call this, the *embedded control group*; and, often, for convenience we will simply call it, the *control group*. The crucial fact is that, within the wreath product, the members of the control group send the fiber-group copies onto each other via *conjugation*. Therefore, we conclude:

Transfer of the fiber-group copies is modeled by their
algebraic conjugation, **within the wreath product,**
by the members of the control group.

Now a crucial role of the New Foundations to Geometry is that they give a comprehensive theory of **raw data representation** such that the data set is maximally reusable in the **data lifecycle**. The book *A Generative Theory of Shape* [17] elaborates this theory in detail and comprehensively explains the use of this theory to represent **scientific data** and **manufacturing data**. The present paper will give a summary of some parts of this theory.

First we should note the following. Given each member c of the control set C, let us call the set-theoretic Cartesian product $F \times \{c\}$ the corresponding **fiber-set copy**, which will also be notated as F_c. By the principle of the Maximization of Transfer, any data set will actually be the union $F \times C$ of the fiber-set copies given by a transfer structure, i.e., a wreath product $G(F)\textcircled{w}G(C)$. Thus, given a wreath product, we will refer to the union of the fiber-set copies as the **data set**. A crucial fact is that there is a group action of the wreath-product group $G(F)\textcircled{w}G(C)$ on the data set $F \times C$ as follows: Given an element in the wreath-product group, i.e., an ordered pair $\langle\ \phi \mid g\ \rangle$ where $\phi \in \prod_{c \in C} G(F)_c, g \in G(C)$, and given an element (f, c) in the data set, define the effect of the former element on the latter, thus:

$$\langle\ \phi \mid g\ \rangle(f,c) \;=\; (\phi(gc)f, gc) \;\in\; F \times C. \tag{2}$$

Notice that this relates the **data-set element** (f, c) in the fiber-set copy F_c to the **data-set element** $(\phi(gc)f, gc)$ in the fiber-set copy F_{gc}.

3 Modularization created by Maximizing Reuse <u>within</u> the Object

This section gives an example to illustrate the Mathematical Theory of Transfer. One of things the example will illustrate to the reader is the **modularization** created by **maximizing the reuse <u>within</u> the object**.

Later in this paper, we will consider much more complex examples. But to enable the reader to begin to understand the mathematical theory, we will initially study a simple example. This example is the way the theory structures a square. We will model the typical way in which a person draws a square on a sheet of paper – i.e., drawing the sides sequentially around the square. Notice that this in fact involves a crucial transfer structure as follows:

The first side is generated by starting with a corner point, and applying translations to trace out the side, as shown in Fig 2.

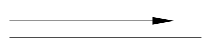

Figure 2: The generation of a side, using translations.

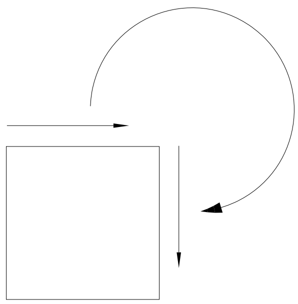

Figure 3: Transfer of translation by rotation.

Next, this translational structure is *transferred* from one side to the next – rotationally around the square. In other words, there is *transfer of translations by rotations*. This is illustrated in Fig 3.

Therefore, according to our theory, the transfer structure, i.e., reuse structure, is defined by the wreath product:

$$\text{Translations} \ ⓦ \ \text{Rotations}$$

where Translations is the fiber group (generating the side) and Rotations is the control group **reusing** the translation program that generated the side. This will now be defined rigorously, as follows:

The translation group will be denoted by the additive group $\mathbb{R}$. The rotation group is $\mathbb{Z}_4$, the cyclic group of order 4, which will be represented as

$$\mathbb{Z}_4 \ = \ \{ \ e, \quad r_{90}, \quad r_{180}, \quad r_{270} \ \}$$

where r_θ means clockwise rotation by θ degrees. We now construct our wreath product of these two groups.

The control group $G(C)$ will be $\mathbb{Z}_4$, and the control set C will be the set of four side-positions around the square:

$$c_1 = \text{top}, \quad c_2 = \text{right}, \quad c_3 = \text{bottom}, \quad c_4 = \text{left}. \tag{3}$$

The effect of the control group $\mathbb{Z}_4$ on the control set $\{c_1, c_2, c_3, c_4\}$ will correspond to the clockwise rotation of the four side-positions onto each other.

The fiber group $G(F)$ will be the translation group $\mathbb{R}$, and the fiber set F will be the infinite line containing the finite side as a subset. The relationship between the infinite line F and the finite side, that it contains, will be defined in our mathematical theory in a crucial way to be described later. First, however, we note that the action of the fiber group $\mathbb{R}$ on the fiber set F will be the obvious translation of the infinite line along itself.

The fact that there are four elements in the control set $\{c_1, c_2, c_3, c_4\}$ implies that there are four fiber-group copies, which will be denoted as $\mathbb{R}_{c_1}, \mathbb{R}_{c_2}, \mathbb{R}_{c_3}, \mathbb{R}_{c_4}$. Also, it implies that there are four fiber-set copies, which will be denoted $F_{c_1}, F_{c_2}, F_{c_3}, F_{c_4}$. These are the four infinite lines that contain the four finite sides as subsets.

It is crucial to understand that each fiber-group copy (translation group $\mathbb{R}_{c_i}$) will act on its own "personal" copy of the fiber set (infinite line F_{c_i}). That is, for each member c_i of the control set, we have the corresponding group action

$$\mathbb{R}_{c_i} \times F_{c_i} \longrightarrow F_{c_i}.$$

Based on this, we can now define the wreath product:

$$\mathbb{R} \circledR{w} \mathbb{Z}_4 \tag{4}$$

First observe that this is the semi-direct product:

$$[\mathbb{R}_{c_1} \times \mathbb{R}_{c_2} \times \mathbb{R}_{c_3} \times \mathbb{R}_{c_4}] \;\circledS_\tau\; \mathbb{Z}_4 \tag{5}$$

where τ, the automorphism representation,

$$\tau : \mathbb{Z}_4 \longrightarrow Aut\{\mathbb{R}_{c_1} \times \mathbb{R}_{c_2} \times \mathbb{R}_{c_3} \times \mathbb{R}_{c_4}\}$$

is such that, given any element in the control group, i.e., a rotation r_θ, its automorphic effect $\tau(r_\theta)$ on the fiber-group product, $\mathbb{R}_{c_1} \times \mathbb{R}_{c_2} \times \mathbb{R}_{c_3} \times \mathbb{R}_{c_4}$, corresponds to the effect of that rotation on the control set $\{c_1, c_2, c_3, c_4\}$. Therefore, the fiber-group copies are rotated around the square; i.e., they are *reused* around the square.

Now let us understand the data set $F \times C$, in this example. It is the *disjoint union* of the four fiber-set copies; i.e., the four infinite lines containing the four finite sides. Therefore it is important to understand that the fiber-set copies are independent sets, i.e., the four infinite lines do not intersect but *overlap*. To help understand this, one can think of them as *four infinite wires* overlapping each other.

Let us now model their relationship to the finite sides. According to the two fundamental principles, Maximization of Transfer, and Maximization of Recoverability, the relationship is this: First, using the Theory of Recoverability of the generative operations (as described later), the four finite sides are generated by cutting down the visibility of the four infinite lines at the end-points of the finite segments, by an extra generative operation that switches the visibility on and off. Furthermore, using the Theory of Transfer, the switching operation is incorporated as follows: First, it is defined by what our theory calls the **occupancy group**, $\mathbb{Z}_2$ (a cyclic group of order 2). The group switches between two states, "occupied" and "non-occupied", which, in the current example, determines whether a point is visible or not visible. Also, by the Maximization of Transfer principle, this group is *transferred* to each point along the infinite line, because the option of switching on and off the side-drawing program is available at any point along the infinite line.

Therefore, by the Mathematical Theory of Transfer, the group $\mathbb{Z}_2$ is placed as a fiber group below the group given in expression (4), thus:

$$\mathbb{Z}_2 \circledR{w} \mathbb{R} \circledR{w} \mathbb{Z}_4.$$

Notice that, with respect to the left wreath product symbol ⓦ, the occupancy group $\mathbb{Z}_2$ is the fiber group, and the subsequence $\mathbb{R}$ ⓦ $\mathbb{Z}_4$ is the control group. Therefore, the subsequence $\mathbb{R}$ ⓦ $\mathbb{Z}_4$ has the effect of *transferring* the occupancy group. Our theory states that transfer maps the fiber-group copies onto each other. In the present case, the fiber-group copies are the copies of the occupancy group, i.e., one copy at each point in the data set $F \times C$ of group $\mathbb{R}$ ⓦ $\mathbb{Z}_4$. Therefore, the copies of the occupancy group can be identified with the points in the data set $F \times C$. Furthermore, since the group $\mathbb{R}$ ⓦ $\mathbb{Z}_4$ transfers the copies of the occupancy group onto each other, we can understand the group $\mathbb{R}$ ⓦ $\mathbb{Z}_4$ as *transferring* the points in the data set onto each other. Therefore, we have this crucial conclusion: There is only one point. The remaining points have been created by *transferring* that point. Therefore, the square was created purely from a single point. The other points are merely transfers of that single point. This is an example of the principle of the Maximization of Transfer.

With respect to notation, we now make the following comment. To help the reader understand the mathematically rich aspects of a structure, such as $\mathbb{R}$ ⓦ $\mathbb{Z}_4$, the occupancy level will usually be omitted from the notation, when it is not needed in the immediate discussion.

The next thing to observe is this: The members of the data set $F \times C$, of the wreath product $\mathbb{R}$ ⓦ $\mathbb{Z}_4$, can be defined *generatively* using the levels of the wreath product. We will now see that the group gives *generative coordinates* to the square, in the following way: The members c of the control set, i.e., the side positions, can be identified with the members r_θ of the control group; where the position of the first side is labeled by the identity element e of the control group, and the positions of the other sides are labeled by the rotations that produced those positions. Thus, any fiber-set copy F_c can be labeled F_{r_θ}. Furthermore, the members f of the fiber set, i.e., the points along a side, can be identified with the translation members t of the fiber group; where the initial point within a side is labeled by the identity element e of the fiber group, and the other points of a side are labeled by the translations that produced those points. Thus, the fiber set F can be identified with the fiber group $\mathbb{R}$. Furthermore, each element in the fiber-set copy $\mathbb{R}_{r_\theta}$ can be labeled t_{r_θ}. As a result of this, any point (f, c) on the square can be described by a pair of coordinates:

$$(t, r_\theta) \;=\; t_{r_\theta} \;\in\; \mathbb{R}_{r_\theta}. \tag{6}$$

That is, in the ordered pair, (t, r_θ), the first coordinate gives the generative (translational) distance along a side, from the side's starting point; and the second coordinate gives the generative (rotational) distance of a side from the first generated side.

This example begins to illustrate the following crucial principle of our theory:

RAW DATA REPRESENTATION. *In the New Foundations to Geometry, every element in a raw data set is defined generatively, such that the generative representation accords with the two basic principles, the Maximization of Transfer and Maximization of Recoverability, as well as the mathematical union of these principles.*

This theory of Raw Data Representation will be illustrated later in this paper, with *complex* examples. However, as an initial illustration, let us return to the particular example we are studying in this section, and observe the following: The four points marked in Fig 4 are four points in the **raw data set** of the points in the square. The generative coordinates which our theory uses to define these coordinates generatively are shown as the ordered pairs at each point. Furthermore, the transfer structure within that generative structure, is also illustrated in the figure. That is, the figure illustrates the fact that, according to the Mathematical Theory of Transfer, the control group element, r_{90}, rotation by 90^{o}, shown by the circular arrow, *transfers* the translation t shown on the top side to the translation t shown on the right side. That is, the translation on the top side is *reused* on the right side. This is incorporated into the coordinates as follows:

Notice that, by the coordinates shown, the starting point for drawing the entire square is the top-left point, because this point is given the generative coordinates (e_1, e_2) where e_1 is the identity element of the fiber group $\mathbb{R}$, and e_2 is the identity element of the control group $\mathbb{Z}_4$. That is, e_1 indicates that this is the starting point along the side, and e_2 indicates that this side is the first side. With respect to our theory of Raw Data Representation, notice that the generative coordinate pair (e_1, e_2), which is our representation of this *raw data* point, defines this raw data point generatively as the identity element of the fiber-group copy $\mathbb{R}_{e_2}$.

Now, consider a translation t along the top side, as shown by the arrow on that side. It moves to the position marked (t, e_2), because the coordinate t indicates that this point is *generated* by the translation t chosen from the fiber group $\mathbb{R}$, and the coordinate e_2 indicates that we are still on the starting side. With respect to our theory of Raw Data Representation, the raw data set point at the second marked position on the top line is given a generative coordinate pair (t, e_2), which, by equation (6), defines this raw data point generatively as the element t_{e_2} of the fiber-group copy $\mathbb{R}_{e_2}$.

Next consider the *transferred* version of this translation on the right side. Within the right side, the start point of this transferred translation is the top right point in the diagram, and is given the generative coordinates (e_1, r_{90}), where e_1 is the identity element of the fiber group $\mathbb{R}$, and r_{90} is the element of the control group $\mathbb{Z}_4$ that generatively relates the position of this side with respect to the starting side. Therefore, e_1 indicates that this is the starting point along the side, and r_{90} indicates that this side is the second side. With respect to our theory of Raw Data Representation, the raw data set point at the first position on the top line is given a generative coordinate pair (e_1, r_{90}), which defines this raw data point generatively as the identity element of the fiber-group copy $\mathbb{R}_{r_{90}}$.

Next, consider the translation t along the right side, as shown by the arrow on that side. It moves to the position marked (t, r_{90}), because the coordinate t indicates that, along this side the point is *generated* by the translation t chosen from the fiber group $\mathbb{R}$; and the coordinate r_{90} indicates that this translation is on the right side. With respect to our theory of Raw Data Representation, the raw data set point at the second marked position on the right side is given a generative coordinate pair (t, r_{90}), which, by equation (6), defines this raw data point generatively as the element $t_{r_{90}}$ of the fiber-group copy $\mathbb{R}_{r_{90}}$.

Now, recall that we saw, on the top side, that our theory represents the second point as a **data set element** by the coordinates (t, e_2) which corresponds to the element t_{e_2} of the fiber-group copy $\mathbb{R}_{e_2}$. Furthermore, we have just seen, on the right side, that our theory represents the second point as a data set element by the coordinates (t, r_{90})

which corresponds to the element $t_{r_{90}}$ of the fiber-group copy $\mathbb{R}_{r_{90}}$. The important thing to understand is that, since r_{90} is a member of the control group, and this defines the fiber-group copy $\mathbb{R}_{r_{90}}$ along the right side as a transferred version of the fiber-group copy $\mathbb{R}_{e_2}$ along the top side – by algebraic conjugation – this represents the fact that the translation $t_{r_{90}}$ used on the right side is a transferred version, i.e., a *reused* version, of the translation t_{e_2} that was used on the top side. Therefore, since the data set representation, (t, r_{90}) and (t, e_2), of these two points corresponds respectively to the two translations $t_{r_{90}}$ and t_{e_2}, the data set representation defines the two points as related by the transfer of their generative history.

As a result of this method, we see the crucial fact that each point, as a data-set element, is given a complete generative description, from the starting point, that maximizes transfer; i.e., maximizes reuse.

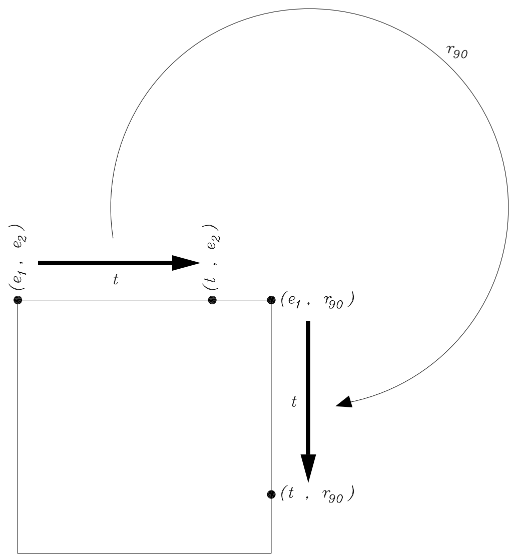

Figure 4: The reuse structure in the generative coordinates.

Let us end this section with the following observation: Recall that the Modularization principle, given on page 5, states that the maximization of transfer of the operations that generate the object has the effect of modularizing the object into generative components that are *maximally reused* within the generation of the object. The example we have been describing clearly illustrates this, as follows: The wreath product $\mathbb{R} \text{ⓦ} \mathbb{Z}_4$, that our theory created to define a square, defines it as generated by the rotation of the translation that generated a side. Therefore, this wreath product *modularizes* the square into its sides, and represents this modularization as the *maximum reuse* of the program that generates a side.

4 Reuse of Reuse

So far, in the example of the square, we have been considering the maximization of reuse *within* the square. Now let us consider the reuse of the square itself. According to our theory, this will mean that the reuse that occurred *within* the square must itself be reused, i.e., there will be a *reuse of reuse*. We will now see how the New Foundations to Geometry mathematically defines this.

As an example, imagine that, in the *data lifecycle*, the square object, which was created in one project, is to be reused in another project. Furthermore, in the latter project, the reuse requires the object to be in a deformed version, e.g., a parallelogram.

This is again handled by the Mathematical Theory of Transfer. According to this theory, the *deformation* of a shape is given by adding an extra layer of transfer, i.e., an extra layer of reuse. For example, to obtain a parallelogram, one adds the group of linear transformations, $GL(2,\mathbb{R})$, onto the two-level group of the square thus:

$$\mathbb{R} \circledw \mathbb{Z}_4 \circledw GL(2,\mathbb{R}). \tag{7}$$

The crucial fact to observe about this expression is that the operation used to add $GL(2,\mathbb{R})$ on to the lower structure $\mathbb{R} \circledw \mathbb{Z}_4$ is, once again, the wreath-product $\circledw$ which means that $GL(2,\mathbb{R})$ acts by *transferring* $\mathbb{R} \circledw \mathbb{Z}_4$, as follows: Since the fiber group $\mathbb{R} \circledw \mathbb{Z}_4$ represents the structure of the square, this means that $GL(2,\mathbb{R})$ transfers the structure of the square onto the parallelogram.

The next crucial fact is that we saw that structure $\mathbb{R} \circledw \mathbb{Z}_4$ of the square is itself a transfer structure; i.e., the transfer of translations by rotations. Therefore, we have the following important consequence:

The transfer structure within the square is itself transferred, by $GL(2,\mathbb{R})$, onto the parallelogram. Diagrammatically, we can illustrate this as follows: Recall the transfer structure shown in Fig 4. According to our mathematics, this is transferred onto Fig 5.

The consequence is this:

REUSE OF REUSE

The above mathematics gives <u>transfer of transfer</u>; i.e., <u>reuse of reuse</u>.

This <u>recursive transfer</u>, i.e., <u>recursive reuse</u>, is encoded by successive wreath product operations $\circledw$.

To illustrate this further, recall from section 3 that there is an extra transfer level below expression (7). That is, an individual side is the transfer of a *point* by translation. The point is given by the occupancy group $\mathbb{Z}_2$. Therefore, the complete structure of the parallelogram is given thus:

$$\mathbb{Z}_2 \circledw \mathbb{R} \circledw \mathbb{Z}_4 \circledw GL(2,\mathbb{R}). \tag{8}$$

What has been illustrated here is the principle of the Maximization of Transfer, i.e., *maximization of reuse*: The parallelogram is given a generative description, all the way up from a point, that maximizes transfer; i.e., maximizes reuse. That is:

The point is transferred by translations to create a side.
The side is transferred by rotations to create a square.
The square is transferred by linear transformations to create a parallelogram.

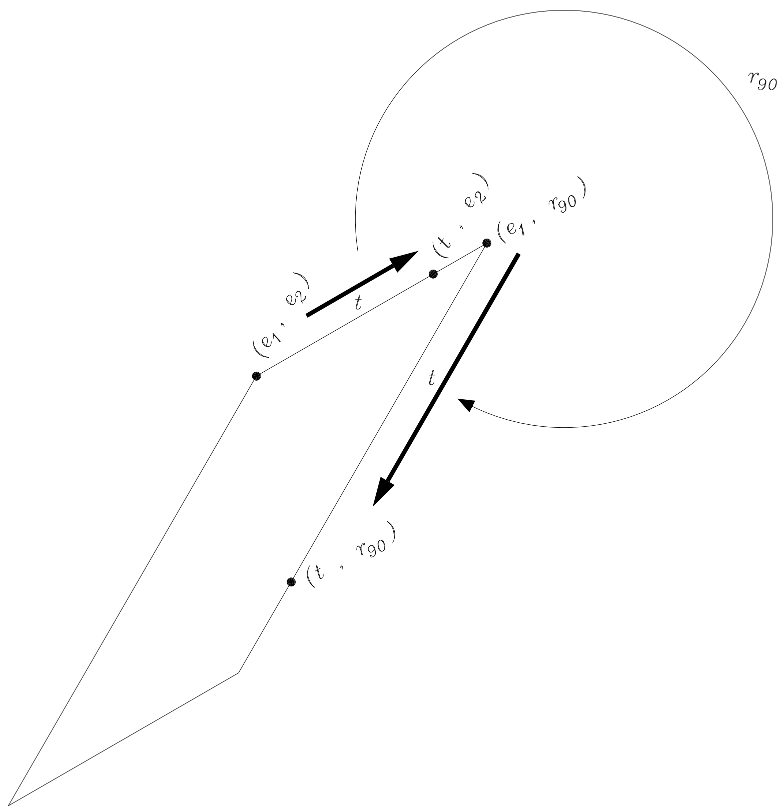

Figure 5: Maximization of transfer has the effect of transferring the transfer structure of the square shown in Fig 4 onto the parallelogram; i.e., transfer of transfer.

5 Persistent Reuse

Persistent reuse is a fundamental requirement in large-scale engineering and scientific systems. Examples, include the following:

> **NASA's Earth Sciences Data Systems has stipulated that reuse of scientific data, and the algorithms that obtained the data, must be extended so that they persist for 1000's of years.**
>
> **In CAD, a serious problem, called the persistent naming problem [3], is the fact that, after feature-editing operations are applied to data in the CAD model, additional features of the data have changed unpredictably and inconsistently with respect to the goal of the applied operations.**

The New Foundations to Geometry solve these persistence problems. Many illustrations will be given in this paper. As an initial illustration, consider again the example we have just been studying.

Consider, in Fig 5, the two translation arrows shown. Notice that, in a *direct drawing* of the parallelogram; i.e., without reuse, the arrow along the top side would give a translation of a different length from the translation given by the arrow on the right side. However, in the reuse of reuse structure, the two arrows define the translations to be the same length as each other, as follows: Consider the linear transformation that transfers the square onto the parallelogram. It also maps the coordinate frame used for the square onto the coordinate frame used for the parallelogram. Therefore, in the coordinate frame for the parallelogram, the two translations we have been considering are the same on the two sides of the parallelogram. This illustrates our principle on page 5 called the Inside-to-Outside Maximization of Reuse, which states that maximization of reuse in the generation of an object maximizes the reusability of the object itself elsewhere. In the current situation, this principle would say this: Maximization of reuse within the square has achieved the reusability of the square's generative operations for the parallelogram because the reuse within the square, i.e., the equal translations within the square, have become, under the linear transformation, the equal translations in the parallelogram. Notice that this would also occur for any subsequent reuse; i.e., persistence of reuse has been achieved.

Now, since the transferring linear transformation is *recovered* from the parallelogram, this illustrates the two basic principles of our Mathematical Theory of Intelligence: Maximization of Transfer and Maximization of Recoverability of the generative operations that created the object. Thus, as illustrated by this example, our theory claims that *persistence* of reuse is achieved by these two principles.

The way maximization of transfer and recoverability is achieved for complex objects will be shown later in the paper. However, again, at this stage, to help the reader understand basic aspects of the Mathematical Theory of Transfer, we will now describe the analysis of another simple object, a cylinder.

We note first that, in computer vision and graphics, cylinders are described generatively as the sweeping of a circular cross-section in the direction of the axis. This can be regarded as defining the cylinder by transfer. However, the group of this sweeping structure has never been given. In contrast, the Mathematical Theory of Transfer creates the following group theory of the structure of the cylinder.

By the principle of the Maximization of Transfer, which the theory claims is fundamental to persistent reuse, we proceed as follows:

Consider first the cross-section. This is given generatively by a circular rotation of a point, as illustrated in Fig 6. That is, it is given by the following structure of *transfer*.

$$\mathbb{Z}_2 \circledR SO(2) \tag{9}$$

i.e., a *single point*, given by the occupancy group $\mathbb{Z}_2$, is transferred by the group $SO(2)$ which is the rotation group in a plane.

Next, the sweeping of the cross-section, in the direction of the rotation axis, is given by the *transfer* of the generative structure of the cross-section by translation, as illustrated in Fig 7. Therefore, the wreath product in expression (9) is given as the fiber-group to which one adds, via an additional wreath product, the translation group $\mathbb{R}$ as the control group, thus:

$$\mathbb{Z}_2 \circledR SO(2) \circledR \mathbb{R}. \tag{10}$$

Notice therefore that, as a result of this structure, the cylinder is decomposed into a structure of generative fiber-group copies, as illustrated in Fig 8.

Now, just as we did in the example of a square, imagine that, in the **data lifecycle**, the cylindrical object, which was created in one project, is to be reused in another project. Furthermore, in the latter project, the reuse requires the object to be in a deformed version, e.g., a bent cylinder.

This is handled in the Mathematical Theory of Transfer, by adding to expression (10), an extra layer of transfer, the deformation group *Diff*, thus:

$$\mathbb{Z}_2 \circledR SO(2) \circledR \mathbb{R} \circledR \mathit{Diff}. \tag{11}$$

The crucial fact to observe about this expression is this: it transfers the structure of the straight cylinder onto the bent cylinder, as illustrated in Fig 9.

Once again, this illustrates the principle of the Maximization of Transfer. According to this theory, the cylinder is generated from only a *single point*, by the **maximal reuse** of the point, thus:

The point is transferred by rotations to create the cross-section.
The cross-section is transferred by translations to create the straight cylinder.
The straight cylinder is transferred by deformations to create a bent cylinder.

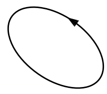

Figure 6: A point is transferred by rotations, producing a circle.

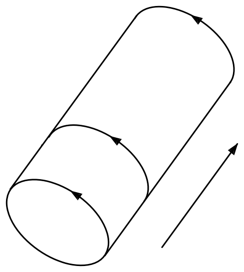

Figure 7: The circle is then transferred by translations, producing a straight cylinder.

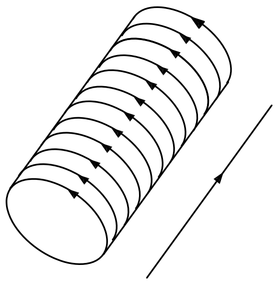

Figure 8: As a result of transfer, a cylinder decomposes into fibers.

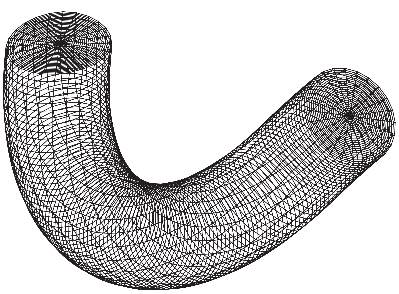

Figure 9: A bent cylinder is a *transfer of the transfer structure* that created the straight cylinder.

Notice that this illustrates again our principle on page 5 called the Inside-to-Outside Maximization of Reuse, which states that maximization of reuse within the generation of an object maximizes the reusability of the object itself elsewhere. That is, in the current example, the reuse structure of the straight cylinder has enabled its reuse as a bent cylinder, as can be seen in Fig 9 which shows that one understands the bent cylinder as a transferred version of the reuse structure of the straight cylinder.

6 Reuse Modularization Notations

According to our theory, reuse modularization is given by a wreath product. It is now valuable to examine the fact that, earlier in this paper, two notations were given for a wreath product. These notations were illustrated with the example of the square, as follows: The first notation uses the symbol ⓦ thus:

Implicit Notation: $\mathbb{R} \text{ ⓦ } \mathbb{Z}_4$

Observe that, to the left of the ⓦ symbol, is the fiber group. Therefore, this notation gives the module structure that is reused. Notice that, even though the group contains four copies of the fiber group $\mathbb{R}$, they are not explicitly shown in the notation. Therefore, while the notation gives the control group and the structure of the module that it reuses, it does not explicitly list the specific instances of the module structure.

Now, recall that, for the *same wreath product group*, there is a notation that uses the semi-direct product symbol ⓢ thus:

Explicit Notation: $[\mathbb{R}_{c_1} \times \mathbb{R}_{c_2} \times \mathbb{R}_{c_3} \times \mathbb{R}_{c_4}] \text{ ⓢ}_\tau \ \mathbb{Z}_4$

Observe that, to the left of the ⓢ symbol, are the fiber-group copies. Therefore, this notation explicitly exhibits the reuse instances of the module structure.

Thus we conclude: The implicit notation gives the control group and the structure of the module that it reuses. The explicit notation gives the control group and the reuse instances of the module structure.

7 Alternative Modularizations

We will now illustrate the crucial fact that, in the Mathematical Theory of Transfer, some situations can be given alternative modularizations that are equally valuable, and are indeed usefully exploited by human beings.

To illustrate this, we will show how the Mathematical Theory of Transfer models the two most popular ways of drawing a square: (1) The first is the sequential drawing of the sides around the square, which corresponds to the transfer structure described earlier. The other most popular scenario is this: (2) First draw the top side followed by the bottom side, each of these two sides individually traced in the left-to-right direction; and then draw the left side followed by the right side, each of these two sides individually traced in the top-to-bottom direction.

What we will now show is that scenario (2) is also chosen because it maximizes transfer, i.e., *maximize reuse*. To demonstrate this, we must consider the reflectional structure of the square. Fig 10 exhibits three of the four reflection axes of the square. They correspond to three reflection operations: reflection m_V about the vertical axis, reflection m_H about the horizontal axis, and reflection m_D about one of the diagonal axes.

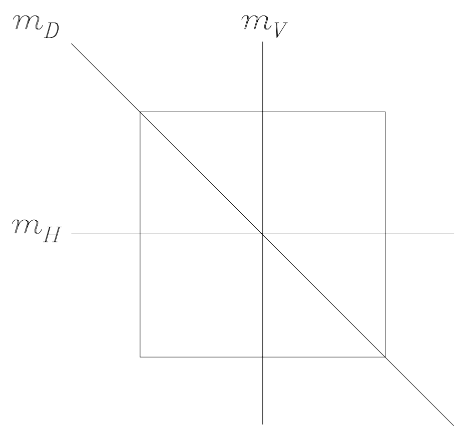

Figure 10: Three of the four reflection operations on a square.

Now observe the following crucial fact: The diagonal reflection operation *transfers* the vertical reflection operation onto the horizontal reflection operation. That is: the diagonal reflection operation describes the horizontal reflection operation as a *reuse* of the vertical reflection operation.

Therefore, in the Mathematical Theory of Transfer, the vertical reflection and horizontal reflection correspond to *fiber-group copies*; and the diagonal reflection corresponds to the *control group* which transfers these fiber-group copies onto each other.

To give this wreath product, let us first give the notation for the reflection groups, thus: The reflection group for the vertical axis is $\mathbb{Z}_2^V = \{\ e,\ m_V\ \}$. The reflection group for the horizontal axis is $\mathbb{Z}_2^H = \{\ e,\ m_H\ \}$. The reflection group for the diagonal axis is $\mathbb{Z}_2^D = \{\ e,\ m_D\ \}$.

Since $\mathbb{Z}_2^V$ and $\mathbb{Z}_2^H$ are the fiber-group copies, and $\mathbb{Z}_2^D$ is the control group, the wreath product, is given as follows, in the explicit notation:

$$[\ \mathbb{Z}_2^V\ \times\ \mathbb{Z}_2^H\ ]\ ⓢ_\tau\ \mathbb{Z}_2^D. \tag{12}$$

Therefore explicitly, this notation says: The control group $\mathbb{Z}_2^D$, to the right of the symbol ⓢ, interchanges the two fiber-group copies $\mathbb{Z}_2^V$ and $\mathbb{Z}_2^H$, shown to the left of the symbol ⓢ; i.e., transfers them onto each other.

Notice that, in the implicit notation, this wreath-product group is written as follows:

$$\mathbb{Z}_2 ⓦ \mathbb{Z}_2.$$

Now, the fact that each side is drawn by translation implies that, below the wreath product group we have just described, one needs to add the translation group $\mathbb{R}$ as a fiber to that group. Thus, one obtains the following wreath product:

$$\mathbb{R}\ ⓦ\ [\ \mathbb{Z}_2^V\ \times\ \mathbb{Z}_2^H\ ]\ ⓢ_\tau\ \mathbb{Z}_2^D \quad = \quad \mathbb{R}\ ⓦ\ \mathbb{Z}_2 ⓦ \mathbb{Z}_2. \tag{13}$$

What we have just described is how the Mathematical Theory of Transfer captures the second popular scenario for drawing a square. Thus, in comparing the first and second popular scenarios, we see this: In the first scenario we described, for drawing a square, the *reuse hierarchy,* given by the wreath product $\mathbb{R}$ ⓦ $\mathbb{Z}_4$, *modularizes* the square into *two levels* of modules. In contrast, in the second scenario we described, the *reuse hierarchy,* given by the wreath product $\mathbb{R}$ ⓦ $\mathbb{Z}_2$ ⓦ $\mathbb{Z}_2$, *modularizes* the square into *three levels* of modules.

8 Mathematical Theory of Object-Linked Inheritance

Inheritance is a crucial property in object-oriented programming. It is the passing of properties from a parent to a child. The child takes on these parent properties, but also adds its own.

It is important to note that there are two major types of inheritance: (1) The type of inheritance that is discussed in all books on object-oriented programming is *class* inheritance, which is an *abstraction* hierarchy. This type is specified in the software text. (2) The other type is not an abstraction hierarchy, but a hierarchy in which objects, created at run-time, are defined by dependencies that are associative references to other objects in the hierarchy. For example, in CAD, a model can be given by a graph of its constituents in which the parent-child links determine which objects must be regenerated when the user decides to change some selected object. That is, the alteration of a property of the selected object (a parent) will necessitate changes in the properties of other objects (its children). The crucial fact is that this type of inheritance is used prolifically in all components of **data lifecycles** and **product lifecycles**.

Section 26 will give our mathematical theory of the first type of inheritance; i.e., class inheritance. The present section gives our mathematical theory of the second type of inheritance. We will call this type of inheritance, *object-linked inheritance*. This is our mathematical theory:

ALGEBRAIC THEORY OF OBJECT-LINKED INHERITANCE. *Object-linked inheritance arises from a wreath product:*

$$Child \; ⓦ \; Parent$$

where $Child$ is the command group of the child, and $Parent$ is the command group of the parent. Thus for a set of n objects that are linked hierarchically from Object 1, the ultimate child, up to Object n, the ultimate parent, the complete transform group of the hierarchy is given by

$$G_1 \; ⓦ \; G_2 \; ⓦ \; \ldots \; ⓦ \; G_n$$

where G_i is the personal transform group of Object i.

As an example, the next section shows how the New Foundations to Geometry mathematically represent kinematics as a structure of <u>reuse</u> and shows how this reuse structure gives the object-linked inheritance.

9 Kinematics

The New Foundations to Geometry give New Foundations to Kinematics. To illustrate this, we will first describe the mathematical theory, invented in these foundations, to describe kinematic structures called *trochoids*, which include *epicycloids* and *hypocycloids*, as examples.

> **Trochoids are kinematic structures that occur for example in <u>planetary gear trains</u> where there is a central gear, called the <u>sun gear</u>, and a gear called a <u>planetary gear</u> rolling around the periphery of the sun gear.**

First, to illustrate a trochoid, consider Fig 11 which shows a circle that is rolling on some other circle. The former circle will be called the roller circle, and the latter circle will be called the pitch circle. Note, in this figure, that the center of the roller circle is labeled O', and the center of the pitch circle is labeled O. An assumption will be that the roller circle is not slipping on the pitch circle. In Fig 11, the left diagram shows the starting state, and the right diagram shows a later state. As illustrated in the right diagram, the angle marked t is the amount that the center O' of the roller circle has rotated about the center O of the pitch circle. Also, as illustrated in the right diagram, the angle marked u is the amount that the roller circle has rotated about its own center O'.

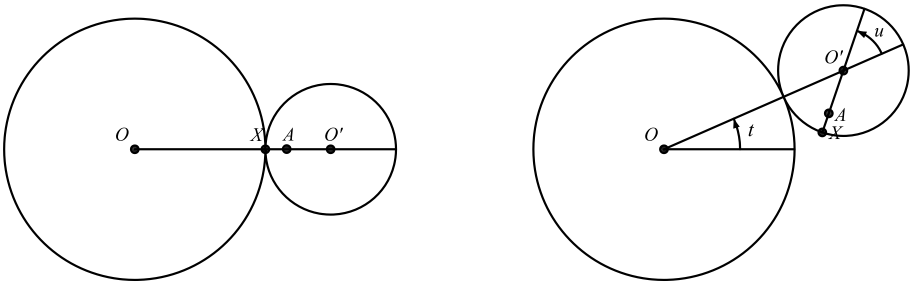

Figure 11: Two states within the creation of a kinematic curve called a trochoid.

A *trochoid* is a curve produced by this kinematic structure as follows. In Fig 11, the point marked A is a chosen point that is fixed relative to the roller circle; i.e., it rotates with the roller circle. Clearly, the point A will trace out a curve. It is such a curve that is called a trochoid. Point A is called the *trace-point*.

Fig 12 shows four examples of trochoids. In each of these four diagrams, the pitch circle is the largest circle, i.e., boundary circle, shown in the diagram, and the roller circle is rolling on the pitch circle *inside* the pitch circle. Also, in each case, the chosen trace point, shown as a large dot, is a fixed point on the edge of the roller circle. Such situations are called *hypocycloids*. When the roller circle is rolling on the pitch circle *outside* the pitch circle, and the trace point is also chosen to be a fixed point on the edge of the roller circle, the situation is called an *epicycloid*. For example, in Fig 11, the point X would give an epicycloid.

Now, in Fig 12, the four hypocycloid curves are the cusp-shaped curves shown in the four diagrams. The four cases shown are as follows: What is responsible for the differences between these four cases is this: The ratio of the radius R of the pitch circle to the radius R' of the roller circle is different in each case. In the top left diagram, the radius R of the pitch circle is 3 times the radius R' of the roller circle. The resulting curve is called a *deltiod*. In the top right diagram, the radius R of the pitch circle is 4 times the radius R' of the roller circle. The resulting curve is called an *astroid*. In the bottom left diagram, the radius R of the pitch circle is 5 times the radius R' of the roller circle. In the bottom right diagram, the radius R of the pitch circle is 6 times the radius R' of the roller circle. For hypercycloids, because the trace point is inside the pitch circle, the ratio R/R' is expressed as a negative number. Therefore, the ratios are the four negative numbers listed in the caption of the diagram.

The next figure, Fig 13, shows two examples of epicycloids. In the left diagram, the radius R of the pitch circle is the same size as the radius R' of the roller circle. The resulting curve is called a *cardioid*. In the right diagram, the radius R of the pitch circle is 2 times the radius R' of the roller circle. The resulting curve is called a *nephroid*.

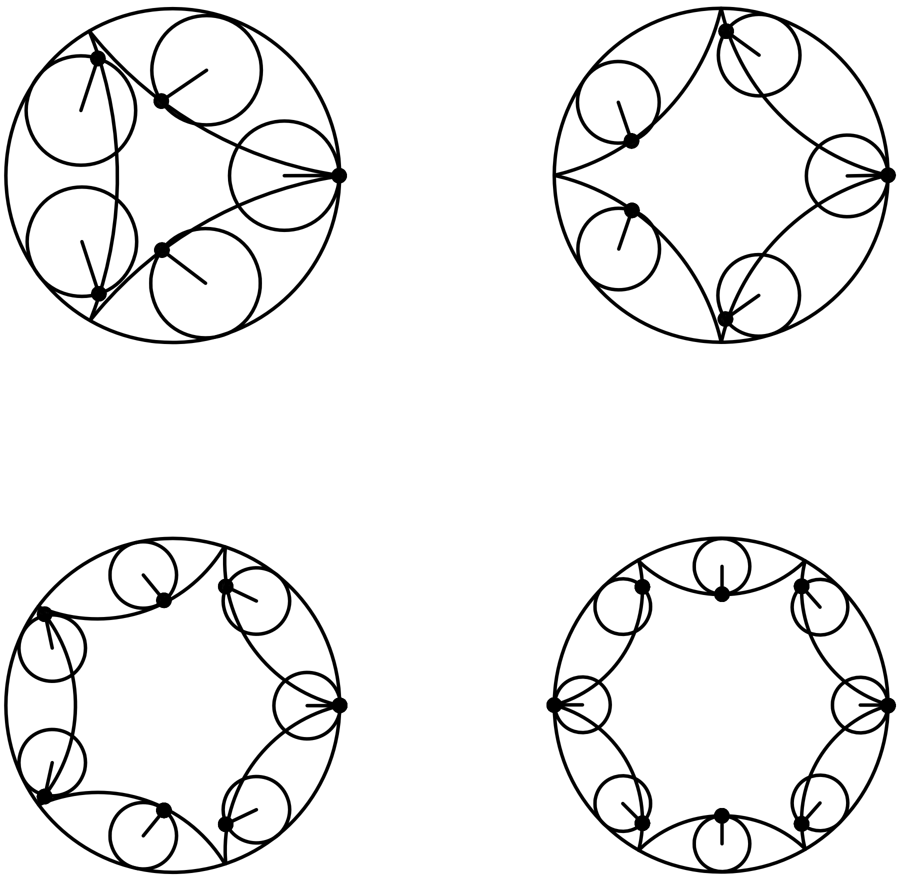

Figure 12: Four examples of hypocycloids, where the ratio of the radius R of the pitch circle to the radius R' of the roller circle is as follows: (top left) $R/R' = -3$, (top right) $R/R' = -4$, (bottom left) $R/R' = -5$, (bottom right) $R/R' = -6$

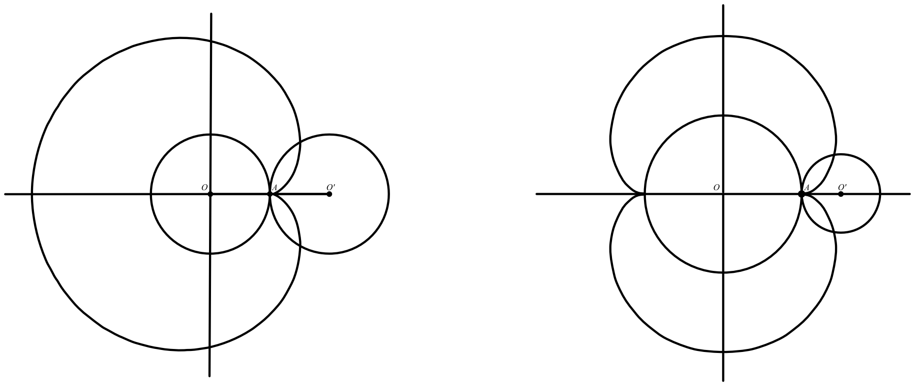

Figure 13: Two examples of epicycloids: (left) $R/R' = 1$ which is called a cardioid; and (right) $R/R' = 2$ which is called a nephroid.

We now claim the following:

> **According to the Mathematical Theory of Transfer, the correct representation of a trochoid shows that it <u>maximizes reuse</u> and is given by the following wreath product:**
>
> $$\mathrm{SO}(2) \textcircled{w} \mathrm{SO}(2) \tag{14}$$
>
> **That is, the rotation group $SO(2)$ transfers the rotation group $SO(2)$. This begins to illustrate our theory that kinematics should be understood as structured by <u>reuse</u>.**

It is important now to understand the relation between the wreath product in (14) and the parameters, as follows: In this wreath product, the control group gives the rotation of the roller-circle center O' about the pitch-circle center O. This rotation corresponds to the increasing angle t in Fig 11. The fiber group gives the rotation of the trace point A about the roller-circle center O'. This rotation corresponds to the increasing angle u in Fig 11.

Notice that the parameter t defines the absolute motion of the line OO' that connects the two centers, and the parameter u defines the trace point's relative motion with respect to that line. Therefore, in the wreath product (14), the control group gives the *absolute motion*, and the fiber group gives the *relative motion*.

This illustrates our algebraic theory of relative motion:

ALGEBRAIC THEORY OF RELATIVE MOTION. *Relative motion follows from the principle of the Maximization of Transfer. To obtain a relative motion representation, apply the following rule:*

> Decompose the motion into two symmetry groups, such that one group transfers the other.

The two groups correspond to the absolute and relative motions respectively. This means that the motion is represented by the following wreath product:

$$\text{relative motion } ⓦ \text{ absolute motion.}$$

Thus, according to this mathematics,
relative motion representation follows from
the maximization of reuse.

Notice that this is also an example of our Mathematical Theory of Object-Linked Inheritance, as given in section 8. That is, the absolute motion corresponds to the parent object, and the relative motion corresponds to the child object.

Now let us return to the wreath product $SO(2)$ ⓦ $SO(2)$ which, we claim, defines the trochoid as a structure that maximizes reuse. It is important to examine this structure in more detail, as follows: First we define the group action given by the control group thus:

$$\left\{ \begin{array}{cccccc} SO(2) & \times & C & \longrightarrow & C \\ (\ r_t & , & t_i \ ) & \longmapsto & r_t t_i = t + t_i \end{array} \right. \tag{15}$$

where the control set C is the set of angles t of the line OO' relative to its starting orientation; and the control group $SO(2)$ is the set of rotations r_t by angles t. The control set can therefore be identified with a circle S^1.

Next, we define the group action given by the fiber group as follows:

$$\left\{ \begin{array}{cccccc} SO(2) & \times & F & \longrightarrow & F \\ (\ r_u & , & u_i \ ) & \longmapsto & r_u u_i = u + u_i \end{array} \right. \tag{16}$$

where the fiber set F is the set of angles u of the line $O'A$ relative to its starting orientation; and the fiber group $SO(2)$ is the set of rotations r_u by angles u. The fiber set can therefore also be identified with a circle S^1.

The fiber-set copies are defined as follows: For each angular position t of OO', that is, for each t in the control set C, define a copy F_t of the fiber set F, where for any $u \in F$, its copy u_t is the angle u around O' at position t and measured from OO' to $O'A$.

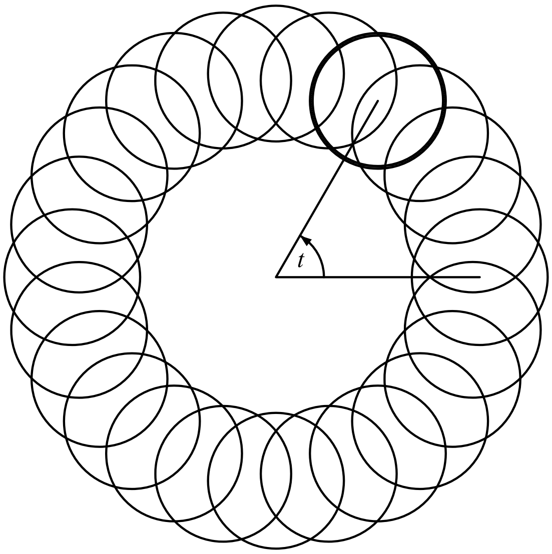

Figure 14: Illustration of the fiber-set copies.

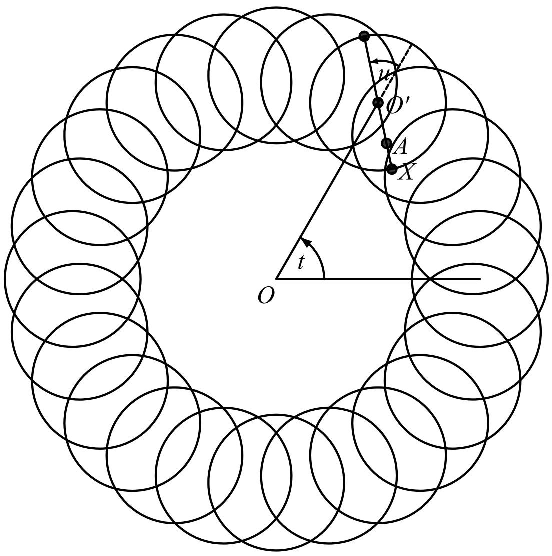

Figure 15: Illustration of an element of the data set.

Fig 14 illustrates the fiber-set copies; i.e., each of the circles in the figure is a fiber-set copy. Note that each circle, i.e., fiber-set copy, is over some member t in the control set, as shown. As we said, t is the angular position of OO' relative to its starting orientation. It will often be convenient to write the fiber-set copy F_t as the set-theoretic Cartesian product $F \times \{t\}$. It is understood as a *copy* of the fiber set because it is located in a particular position that is different from each of the other fiber-set copies; i.e., each fiber-set copy is located in its own place.

Now recall that we define the **data set** of a wreath product to be the disjoint union of all its fiber-set copies, i.e., the set-theoretic Cartesian product $F \times C$. Therefore, in the present case, each element of the data set is of the form

$$(\ u \ , \ t \ )$$

Therefore, Fig 15 illustrates an example of a **data-set element**.

Let us now study the fiber-*group* copies. For each angular position t of OO', that is, for each t in the control set C, define a copy $SO(2)_t$ of the fiber group $SO(2)$, where the members of $SO(2)_t$ are rotations $(r_u)_t$ around the point O'.

To illustrate the fiber-group copies, return to Fig 14. Each fiber-group copy $SO(2)_t$ acts on its corresponding fiber-set copy F_t. Therefore, each fiber-set copy F_t, i.e., each circle in Fig 14, has a fiber-group copy rotating around its center. It is understood as a *copy* of the fiber group because it is located in a particular position that is different from each of the other fiber-group copies; i.e., each copy rotates around a different center. Nevertheless, by our theory, it is a **reuse** of the other fiber-group copies.

We now describe the next crucial step in our Mathematical Theory of Kinematics. In order to illustrate this step in the current example, the reader should observe this: The group we have described so far, in this example, allows the rotation of the roller circle, about its center, to be of any angle u, and therefore allows slipping of the roller circle. Now, as stated earlier, the trochoid has the condition that the roller circle does not slip. This is crucial for example when the trochoid is modeling a planetary gear train. This condition is fulfilled in our Mathematical Theory of Kinematics in the following way:

> **According to the New Foundations to Geometry, imposing the no-slipping constraint for the trochoid is achieved by locating, within the wreath product $SO(2) \circledw SO(2)$, the subgroup $\mathfrak{Gen}_{\mathfrak{T}}$ of elements that generate the trochoid curve. This will be called the trochoid's generative group. That is, the generative group is a subgroup of the reuse structure.**

To locate this subgroup, first note that the no-slipping constraint corresponds to the condition that the arc along which the contact point moves around the pitch circle is equal in length to the arc along which the contact point moves around the roller circle; that is, $Rt = R'u$.

The consequence is that, in the data set $F \times C$, the elements corresponding to the trochoid states are of the form

$$(\, Rt/R' \, , \, t \,). \tag{17}$$

This allows us to locate the generative group $\mathfrak{Gen}_{\mathfrak{T}}$ in the wreath product $SO(2) \circledw SO(2)$, as follows: First note that this wreath product is the following semi-direct product:

$$\{ \prod_{t \in C} SO(2)_t \} \circledS_{\tau} SO(2)$$

Next, given a member r_u of the fiber-group, define, in the fiber-group product, the element that is the *diagonal* element corresponding to r_u to be given by this function:

$$\Delta_{r_u} : \begin{cases} C & \longrightarrow \quad \bigcup_{t \in C} SO(2)_t \\ \\ t & \longmapsto \quad\quad (r_u)_t \end{cases} \tag{18}$$

We shall now see that the trajectory of the trochoid is generated by those members of the wreath product $SO(2) \circledw SO(2)$ that are of the form

$$\langle \, \Delta_{r_{Rt/R'}} \mid r_t \, \rangle \qquad \forall t \in C. \tag{19}$$

This is shown simply by applying any such element to the initial state $(u, t) = (0, 0)$, thus:

$$\langle \, \Delta_{r_{Rt_i/R'}} \mid r_{t_i} \, \rangle \, (0,0) \; = \; (r_{Rt_i/R'})_{t_i} \, (0, t_i) \;\; = \;\; (\, Rt_i/R' \, , \, t_i \,) \tag{20}$$

where we see that the point given by the right-most ordered pair in this expression fits the constraint (17). This means that the trochoid curve is the following subset of elements in the data set

$$\mathfrak{T} : \begin{cases} C & \longrightarrow \qquad\qquad F \times C \\ \\ t_i & \longmapsto \quad \langle \, \Delta_{r_{Rt_i/R'}} \mid r_{t_i} \, \rangle \, (0,0) \end{cases} \tag{21}$$

Furthermore, the elements of the form (19) send the data-set subset $\mathfrak{T}$ to itself, which can be seen as follows: Taking the resulting point in (20), and applying, to it, another element $\langle \, \Delta_{r_{Rt_j/R'}} \mid r_{t_j} \, \rangle$ of the form (19), we get:

$$\begin{aligned} \langle \, \Delta_{r_{Rt_j/R'}} \mid r_{t_j} \, \rangle \, (\, Rt_i/R' \, , \, t_i \,) \;\; &= \;\; (\, \Delta_{r_{Rt_j/R'}}(t_j + t_i)[Rt_i/R'] \, , \, t_j + t_i \,) \\ &= \;\; (\, r_{Rt_j/R'}[Rt_i/R'] \, , \, t_j + t_i \,) \\ &= \;\; (\, Rt_j/R' + Rt_i/R' \, , \, t_j + t_i \,) \\ &= \;\; (\, R(t_j + t_i)/R' \, , \, t_j + t_i \,) \end{aligned} \tag{22}$$

which again fits the no-slipping constraint (17).

We can use the wreath-product group binary operation to precompose the two group elements, from (20) and (22), first, as follows:

$$\begin{aligned}
&\langle\, \Delta_{r_{Rt_j/R'}} \mid r_{t_j} \,\rangle \langle\, \Delta_{r_{Rt_i/R'}} \mid r_{t_i} \,\rangle\, (\, 0\, ,\, 0\,) \\
= \quad &\langle\, \Delta_{r_{Rt_j/R'}} \tau(r_{t_j})[\Delta_{r_{Rt_i/R'}}] \mid r_{t_j} r_{t_i} \,\rangle\, (\, 0\, ,\, 0\,) \\
= \quad &(\, \Delta_{r_{Rt_j/R'}} \tau(r_{t_j})[\Delta_{r_{Rt_i/R'}}](t_j + t_i)0\, ,\, t_j + t_i\,) \\
= \quad &(\, \Delta_{r_{Rt_j/R'}} (t_j + t_i) \tau(r_{t_j})[\Delta_{r_{Rt_i/R'}}](t_j + t_i)0\, ,\, t_j + t_i\,) \\
= \quad &(\, r_{Rt_j/R'} r_{Rt_i/R'} 0\, ,\, t_j + t_i\,) \\
= \quad &(\, Rt_j/R' + Rt_i/R'\, ,\, t_j + t_i\,) \\
= \quad &(\, R(t_j + t_i)/R'\, ,\, t_j + t_i\,) \qquad (23)
\end{aligned}$$

which gives the same result as (22).

From this, we draw the following conclusions: First, the elements (19) form a group. That is, closure is satisfied thus:

$$\begin{aligned}
\langle\, \Delta_{r_{Rt_j/R'}} \mid r_{t_j} \,\rangle \langle\, \Delta_{r_{Rt_i/R'}} \mid r_{t_i} \,\rangle \quad &= \quad \langle\, \Delta_{r_{Rt_j/R'}} \tau(r_{t_j})[\Delta_{r_{Rt_i/R'}}] \mid r_{t_j} r_{t_i} \,\rangle \\
&= \quad \langle\, \Delta_{r_{R(t_j+t_i)/R'}} \mid r_{t_j+t_i} \,\rangle \qquad (24)
\end{aligned}$$

The identity element of this group is $\langle\, \mathcal{E} \mid r_0 \,\rangle$, where $\mathcal{E}$ is the identity element of the fiber-group product. The inverse of an element $\langle\, \Delta_{r_{Rt/R'}} \mid r_t \,\rangle$ is $\langle\, \Delta_{r_{-Rt/R'}} \mid r_{-t} \,\rangle$ as can be seen from (24). It is this group that we denote as $\mathfrak{Gen}_{\mathfrak{T}}$.

Second, the group $\mathfrak{Gen}_{\mathfrak{T}}$ defines a group action on the data-set subset $\mathfrak{T}$ which is the trochoid curve. This is seen simply by the fact that applying two elements from the group successively, as in (20) and (22), gives the same result as composing those two group elements first and then applying the result, as in (23).

In conclusion, we see this:

GENERATIVE GROUP OF TROCHOID: *The generative group $\mathfrak{Gen}_{\mathfrak{T}}$ of a trochoid is the set of all elements $\langle\, \Delta_{r_{Rt/R'}} \mid r_t \,\rangle$ in the wreath product $SO(2)$ ⓦ $SO(2)$, using every $t \in C$.*

The purpose of giving the trochoid example has been to illustrate the techniques used in the Mathematical Theory Kinematics given by the New Foundations to Geometry. The Mathematical Theory is given as follows:

MATHEMATICAL THEORY OF KINEMATICS. *Define the structure of the variables as a wreath product that maximizes transfer, i.e., maximizes* reuse. *Use the kinematic constraints, defining the trajectory, to locate, within that wreath product, the subgroup of group elements that generate the trajectory. The consequence is that the defined generative group of the trajectory exploits the* reuse *structure given by the wreath product.*

The fact that the resulting generative group exploits the reuse structure given by the wreath product was illustrated by the several *algebraic equations* we presented in describing the trochoid generative group.

Now, in order to help the reader understand the algebraic equations, we restricted the trochoid wreath product to two levels. In fact, in our mathematical theory, to *maximize* transfer, there is an extra wreath product level, which will now be described. The value of describing this is that it will show that the New Foundations to Geometry give the mathematics of **planetary motion**. As we stated, when used in mechanical engineering, the trochoid is frequently called the *planetary gear train*. The reason is that it does have certain properties that are similar to planetary motion.

First observe that standardly, in the computer simulation of a planetary motion, two frames are centered at the pivot point (center) of the *parent* object, e.g., the sun. One of the frames F_1 is fixed with respect to the parent object, and the other frame F_2, called the *dummy frame*, rotates in alignment with the child planet that is orbiting around the parent planet. Although the dummy frame is centered in the parent planet, it actually "belongs" to the child planet in the sense that it carries the orbiting relative motion of the child with respect to the parent.

Now, if the child planet is spinning around its own center, as was illustrated in the case of the roller circle of the trochoid, and occurs also in the case of the **planet earth**, there is a translation that maps the dummy frame F_2 out to the pivot point of the child planet, thus giving a frame F_3 centered in the child planet and parallel to the dummy frame in the parent planet. Then the spinning of the child planet is given by a rotation of frame F_3, which results as a frame F_4, which is fixed relative to the child planet.

According to the New Foundations to Geometry, the structure we have just described is an example of our principle of the Maximization of Transfer, and is given by the following three-level wreath product:

$$SO(2) \circledS \mathbb{R} \circledS SO(2)$$

where the $\mathbb{R}$ level has been inserted as an extra wreath level between the two levels of the wreath product $SO(2) \circledS SO(2)$ given earlier for the trochoid. In fact, the group levels, in the three-level wreath product, define the relation between the four successive frames as follows:

$${}_{F_4}SO(2)^{F_3} \circledS {}_{F_3}\mathbb{R}^{F_2} \circledS {}_{F_2}SO(2)^{F_1}$$

Notice that this is an example of the Mathematical Theory of Object-Linked Inheritance as given in section 8. In the case of the trochoid, this wreath product gives the full relation between the spinning roller circle and the pitch circle.

Now, as we said, we omitted the middle level $\mathbb{R}$, in our original discussion of the trochoid, in order to make it easier for the reader to understand the wreath product properties of the algebraic equations.

A further reason for omitting the middle level was this: The amount of translation is constant for the entire orbit of the roller circle. Therefore, it would not have added that much extra information to the algebraic structure.

Let us now turn to the planetary motion of the earth. In this case, because the distance from the sun to the earth is varying along the orbit, the amount of translation between frame F_2 and frame F_3 varies. Therefore, in the case of the earth's planetary motion, it is necessary to give the *full* three-level structure $SO(2) \circledwedge \mathbb{R} \circledwedge SO(2)$.

Notice that, by this mathematical theory, the translation group is *transferred* around the earth's orbit; i.e., *reused*. That is, despite the fact that the translation is varying during the orbit, the group itself is transferred, and the variation of the selected group element is given by the choice of a subgroup, by the type of subgroup technique that we illustrated earlier.

10 Recoverability

Recall that the New Foundations to Geometry are founded on *two* principles: the Maximization of Transfer, and the Maximization of Recoverability. So far in this paper, we have been dealing with transfer, and we now turn to recoverability. Furthermore, we will see that the New Foundations to Geometry unite the theory of transfer and the theory of recoverability into an extremely powerful mathematical structure.

Now, in the New Foundations to Geometry, *recoverability* enters as follows: A purpose of the New Foundations to Geometry is to give a generative representation for every object. To do so, the New Foundations provide the laws and inference rules, by which the generative operations, that created the object, can be recovered from the presented state of the object.

My first book, in MIT Press, was a 630-page analysis of this problem. As a result of that lengthy analysis, one of the fundamental laws that the book proposed and fully demonstrated was this:

ASYMMETRY PRINCIPLE. *The only recoverable operations are symmetry-breaking ones. That is, a generative sequence is recoverable only if it is symmetry-breaking on each of the successively generated states.*

Now consider the fact that there are many processes in the world that are not symmetry-breaking, but are symmetry-increasing; e.g., a tank of gas settling to equilibrium under the standard entropy-increasing process. Concerning recoverability of such processes, the New Foundations to Geometry say this:

SYMMETRY-INCREASING PROCESSES. *A symmetry-increasing process is recoverable only if it is symmetry-decreasing on successive data sets.*

Therefore, as an example, you can recover the fact that the tank of gas was entropy-increasing over time, if you kept a set of records (e.g., photographs) and the records are linearly ordered, e.g., they are laid out from left to right on a table, in which case the sequence of photographs breaks the left-right symmetry of the table. That is, the increase in spatial symmetry in the tank of gas is made to correspond to a decrease in spatial symmetry of the record structure.

Now, before we elaborate the rules of recoverability, it is valuable first to show how the New Foundations to Geometry combine transfer and recoverability.

11 Fundamental Issue: Combining Transfer and Recoverability

The above discussion leads to the following crucial condition: In order to ensure recoverability, the control group must be symmetry-breaking on its fiber. The conclusion is therefore this:

MATHEMATICAL THEORY OF
COMBINING TRANSFER AND RECOVERABILITY

1. Transfer is given by a wreath product.

2. Recoverability adds the condition that, within the wreath product, the control group must be symmetry-breaking on its fiber.

3. Therefore, the combination of transfer and recoverability leads to a powerful mathematical structure invented in the New Foundations called:

<u>symmetry-breaking wreath products</u>.

12 New Theory of Symmetry-Breaking

We will now see that this gives a far more powerful theory of symmetry-breaking than the conventional one that underlies physics and chemistry.

CONVENTIONAL THEORY OF SYMMETRY-BREAKING. *Symmetry-breaking is a reduction of symmetry group.*

To illustrate this, let us consider the transition from a square to a parallelogram. Observe that this is a symmetry-breaking transition, and it looses most of the symmetries in the square. In the conventional view, a square is given by the symmetry group D_4 which consists of the eight Euclidean transformations that map the square to itself: four rotations and four reflections. In contrast, a parallelogram is given by the symmetry group $\mathbb{Z}_2$ which consists of the only two Euclidean transformations that map a parallelogram to itself: rotation by 0^0 and rotation by 180^0. Therefore, in the conventional view, the transition from a square to a parallelogram is given by the following transition of groups:

$$D_4 \quad \longrightarrow \quad \mathbb{Z}_2. \tag{25}$$

Observe that the group $\mathbb{Z}_2$ is in fact a subgroup of D_4. The consequence is that the symmetry-breaking transition is being described as the reduction of a symmetry group.

This view of symmetry-breaking has dominated physics and chemistry for nearly a century.

Now, according to the New Foundations to Geometry, the conventional view is *inherently weak* for a number of major reasons: First, because, in the conventional view, symmetry-breaking is given by a reduction in algebraic structure, there is the following bad consequence: As one goes from a simpler object (such as a square) to a more complex object (such as a parallelogram), one is reducing the size of the description – which is logically absurd. Another bad consequence is that one is loosing information about the past.

We will now describe the far more powerful theory of symmetry-breaking given by the New Foundations to Geometry. To illustrate, recall section 2 that showed how the New Foundations mathematically describe the transition from a square to a parallelogram. The description is this: One takes the symmetry group of the square, which the theory gives as $\mathbb{R} \ⓦ \mathbb{Z}_4$, and *adds* the group of linear transformations, *via a wreath product*. That is:

$$\mathbb{R} \ⓦ \mathbb{Z}_4 \quad \longrightarrow \quad \mathbb{R} \ⓦ \mathbb{Z}_4 \ⓦ GL(2, \mathbb{R}). \tag{26}$$

Therefore, in the New Foundations, symmetry-breaking actually preserves the original group, and in fact increases it. The enormous power of this approach will be seen shortly, but first let us state it precisely:

NEW THEORY OF SYMMETRY-BREAKING

DEFINES

SYMMETRY-BREAKING AS REUSE

The New Foundations to Geometry give the following mathematical theory of symmetry-breaking:

The breaking of a symmetry group G_1 is given by its extension by another group G_2 via a wreath product thus: $G_1 ⓦ G_2$, where G_2 is the symmetry group of the asymmetrizing action.

Therefore, in this theory, symmetry-breaking is the <u>reuse</u> of the past symmetry group.

It is important to carefully understand this theory, as follows: Notice that the wreath product symbol ⓦ in the above statement implies that the *past symmetry G_1 is transferred onto the present broken symmetry*. To illustrate the extreme relevance of the concept, consider the following example: Consider a bent pipe that one might come across in the street. Merely by the fact that one understands it as a bent pipe means that one sees the past symmetric version as transferred onto the present asymmetric version. Therefore, the present structure of the pipe is defined in terms of its history. That is, in accord with the New Foundations to Geometry, the present state of the object is defined by recoverable generative operations.

Now note the following: the image of the bent pipe as *raw data* does not contain the symmetry of the past state, the straight pipe. However, by applying the new theory of symmetry-breaking to this raw data, the data is converted in to a *causal representation* of the pipe, i.e., that it is the result of bending. The consequence is that the raw data has been given a useful representation with respect to scientific inquiry, since science concerns causal explanation. This issue therefore fundamentally relates to Science Data Systems, as follows:

FUNDAMENTAL IMPORTANCE OF

THE NEW THEORY OF SYMMETRY-BREAKING FOR

SCIENCE DATA SYSTEMS (SDS)

The Conventional Theory of Symmetry-Breaking is bad for the needs of Science Data Systems because it represents the data as a reduced version of the past symmetry group, and therefore looses the past state of the object as well as the object's history.

In contrast, the Theory of Symmetry-Breaking given by the New Foundations to Geometry is good for the needs of Science Data Systems because it represents the data in terms of a recovered generative process that transfers the past symmetry onto the current state, and therefore gains the past state of the object as well as the object's history.

Therefore, because this defines the object in terms of the causal processes that produced it, the data is converted into a form that is useful for scientific research, as required by Science Data Systems.

Furthermore, in this new theory of symmetry-breaking, the raw science data is given a structure that is a <u>reuse</u> of past structure. That is, since the past symmetry is not lost, but is transferred onto the present asymmetric state, this means that the present builds on the past, rather than throwing the past away, and the past state is <u>reused</u>.

13 Mathematical Theory of Memory Storage

Data Archives are fundamentally important for Science Data Systems. For example: NASA Earth Science Data System (ESDS) processes and produces Earth science *data products* that are *archived* by Distributed Active Archive Centers (DAACs).

DATA ARCHIVES

Because the New Foundations to Geometry are based on the Maximization of Recoverability principle, the data representations are given a format which optimizes their capacity for archiving.

The recoverability, from an object, of the processes that generated it, means that the object acts as a *memory store* for the processes. In fact, a basic principle of the New Foundations to Geometry is this:

The New Foundations to Geometry give the following mathematical result: Shape is equivalent to Memory Storage.

This is fundamentally different from the Standard Foundations to Geometry, that have existed for 3000 years. The Standard Foundations are based on what is called the *invariance program*, which defines a geometric object as an invariant under actions; i.e., a property that does not change under actions. I have argued that this is fundamentally wrong for the computational age. The crucial resource of the computational age is memory storage, and I have argued that an invariant of actions is *memoryless* with respect to those actions; thus making the Standard Foundations highly inappropriate for the computational age. In contrast, in the New Foundations, the principle of Maximization of Recoverability defines a geometric object as a memory store of the applied actions. Thus:

Whereas the objects of the Standard Foundations to Geometry are memoryless objects (i.e., invariants), the objects of the New Foundations to Geometry are memory stores.

Furthermore, because the New Foundations are based not only on the principle of Maximization of Recoverability, but also on the principle of Maximization of Transfer; and because the New Foundations give the mathematics of recoverability and the mathematics of transfer, and combine those two mathematical systems, there is the following result:

FUNDAMENTAL STRUCTURE OF MEMORY STORES

According to the New Foundations to Geometry:
Any memory store is structured as a
symmetry-breaking wreath product.

An extensive aspect of the New Foundations is that it is the only system that gives a comprehensive mathematical theory of memory storage. This gives a crucial understanding of how objects should be structured for archives in large-scale scientific and engineering systems, in fact, for entire product lifecycles and data lifecycles. The integration of heterogenous lifecycle components requires that the objects of those components, whether data products or engineering designs, allow those objects to be *archives* of the actions that produced them. The above principle shows how they can be structured such that they can act as optimal archives of those actions.

The next thing to observe is this: The fact that, in the New Foundations to Geometry, the data representations are based simultaneously on the mathematics of recoverability and the mathematics of transfer, leads to the following result:

THE NEW FOUNDATIONS TO GEOMETRY

BRING

REUSE INTO THE STRUCTURE OF MEMORY STORES

14 Externalization Principle

We will now go into more detail of the Mathematical Theory of Memory Storage given by the New Foundations to Geometry. Before going into the mathematics, it is first necessary to understand the following distinction made by the New Foundations:

EXTERNAL INFERENCE: *External inference will also be called the* **single-state assumption***. In this type of inference, the observer assumes that the data set contains a record of only a single state of the generative process (i.e., a single snap-shot). Any inferred previous state therefore does not have a record in the data set. Therefore, we will say that the inferred previous state is* **external** *to the data set.*

An example of external inference is this: Suppose one is presented with an object in only a single state, where this state is a *deformed* version of the object; e.g., a bent pipe. The inference that the object had undergone a deforming process, e.g., bending, and originated from a non-deformed version, e.g., a straight pipe, is an *external* inference because the deforming process and past state are external to the data set, which, in this example, contains the record of only one state.

INTERNAL INFERENCE: *Internal inference will also be called the* **multiple-state assumption**. *In this type of inference, the observer assumes that the data set contains records of multiple states of the generative process (i.e., multiple snap-shots taken over time). A state, recorded in the data set, can therefore have a past state that also has a record* **internal** *to the data set.*

An example of internal inference is this: Suppose one is presented with a trace of states, e.g., a track in snow. The inference that the points along the track were produced at different times, and were produced by a succession of movements, are given by *internal* inference because the successive states of the tracing process have records within the data set. We shall often use the phrase *internal structure* and *trace structure* interchangeably.

Now, according to the New Foundations, although external and internal inference are applied in very different types of situations, e.g., deformation vs. traces, they are both carried out by using the Asymmetry Principle (section 10) which states that the generative sequence is recoverable only if it is symmetry-breaking on each of the successively generated states. The application of the Asymmetry Principle, in the two cases, differs with respect to the assumed asymmetries as follows:

EXTERNAL AND INTERNAL APPLICATION: *In external inference, the Asymmetry Principle is applied to asymmetries that are assumed to be* **intra-record** *, i.e.,* **within** *the assumed single state in the data set. In internal inference, the Asymmetry Principle is applied to asymmetries that are assumed to be* **inter-record** *, i.e.,* **between** *the assumed multiple states in the data set.*

A rule will now be stated that turns out to be fundamental to the entire process of recovering generative history. In fact, the rule has enormous universal validation, as follows:

THE MOST POWERFUL LAW OF SCIENCE

The New Foundations to Geometry have proposed a law called the Externalization Principle and demonstrated that this law is universally true of all disciplines, including general relativity, quantum mechanics, biology, computer-aided design and manufacturing, computer vision, music, etc. Furthermore, the New Foundations show that it is the <u>fundamental law</u> of all disciplines.

EXTERNALIZATION PRINCIPLE

> **To maximize recoverability, any generative sequence, inferred by external inference, must lead back to a starting state whose internal structure corresponds to an iso-regular group.**

An iso-regular group is one of the groups invented in the New Foundations to Geometry, and is defined as follows:

ISO-REGULAR GROUP. *This is a group satisfying the following three conditions:*

> *(1) It is an n-fold wreath product, $G_1 \circledS G_2 \circledS \ldots \circledS G_n$; i.e., a structure of hierarchical transfer.*
> *(2) Each level G_i is generated by a single generator (i.e., it is either a cyclic group or a connected 1-parameter Lie group).*
> *(3) Each level G_i is an isometry group on its space of action.*

We will give complex examples of this principle later in the paper, but at first, so that the reader begins to understand it, we give some simple examples.

First, to illustrate the definition of an *iso-regular group*, observe that two of the groups given earlier are examples of iso-regular groups.

Square: $\mathbb{R} \circledS \mathbb{Z}_4$

Cylinder: $SO(2) \circledS \mathbb{R}$

Notice, as required by condition (2) of an iso-regular group, each of the levels, in these two groups, can be generated by a single generator. Furthermore, as required by condition (3) of an iso-regular group, each of the levels acts as an isometry group on its space of action; i.e., preserving the metric on that space.

Before we give complex examples of the Externalization Principle, let us help the reader by first giving a simple example. This example shows that the Externalization Principle is important also in human visual perception.

Fig 16 illustrates the results from a series of psychological experiments I carried out in the 1980s (Leyton [11] [12]). The experiments showed that, when subjects are presented with the first figure in Fig 16, i.e., the rotated parallelogram (Fig 16a), their minds go through the sequence of shapes in the rest of the figure; i.e., they reference the rotated parallelogram to a non-rotated one, which they then reference to a rectangle, which they then reference to a square. It was shown that what their minds are doing is recovering the history of the rotated parallelogram; i.e., conjecturing that previously it was non-rotated, and previous to that it was a rectangle, and previous to that it was a square.

The crucial fact is that the only data that the subjects were actually presented with was the first figure, the rotated parallelogram. Therefore, each of the successive inferences

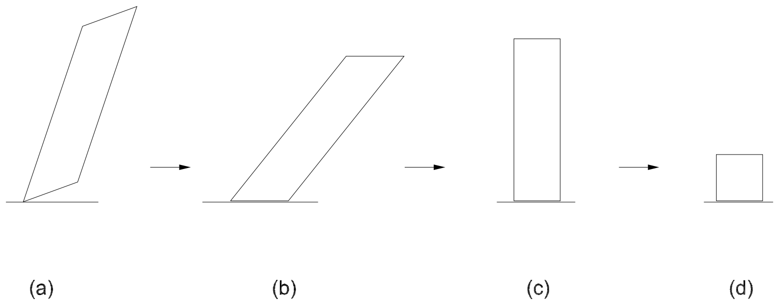

Figure 16: Psychological results found in Leyton [11] [12].

was *external*; i.e., each inferred previous state was external to the presented data – the rotated parallelogram.

It is important now to observe the following: The subjects recovered the successive shapes, backward in time, by successively using the Asymmetry Principle at each stage. That is, the presented figure, the rotated parallelogram, has three asymmetries (distinguishabilities):

> (1) The distinguishability between the orientation of the parallelogram and the orientation of the environment, as indicated for example by the distinguishability between the orientation of the bottom side of the parallelogram and the ground-line.
> (2) The distinguishability in size between adjacent angles.
> (3) The distinguishability in length between adjacent sides.

One can see that the subjects were producing the sequence of shapes by removing these three distinguishabilities successively, backward in time. Notice that this means that, in the *forward-time direction*, i.e., from right to left, the successive stages were symmetry-breaking, in accord with the Asymmetry Principle, which, as stated in (section 10), says that a generative sequence is recoverable only if it is symmetry-breaking on each of the successively generated states.

What is crucial now to notice is that the subjects were using the Externalization Principle. That is, as noted above, the successive states, backwards in time (from left to right), were inferred by *external inference*. Furthermore, this inference goes back to a square – which we have seen has an internal structure given by an iso-regular group. Thus, we see that the sequence accords with the Externalization Principle, which says that external inference always leads back to an internal structure given by an iso-regular group.

To give another example of the Externalization Principle, consider a situation described earlier. Imagine coming across a bent pipe in a road. The fact that one represents it as bent, means that one infers that its generative origin was a straight pipe, i.e., a straight cylinder. This inference is *external* because the straight pipe is not visible in the current situation. Most crucially, observe that the inference accords with the Externalization Principle because the inferred past state, the straight cylinder, is given by an iso-regular group.

15 General Relativity

We will now show that the Externalization Principle of the New Foundations to Geometry is fundamental to general relativity.

According to general relativity, in a gravitational field, the velocity vector of a free falling particle undergoes no change with respect to parallel transport along the particle's trajectory; i.e., the trajectory is a geodesic. Therefore, covariant differentiation along the trajectory of a *single* particle cannot be used to infer the presence of the gravitational field.

However, the presence of a gravitational field can be inferred from *two* particles, in the following way:

First note that, according to differential geometry, the curvature tensor on *curved* manifolds determines that pairs of geodesics which start out parallel do not remain parallel; i.e., there is geodesic deviation. In contrast, on a flat manifold, geodesics that start out parallel remain parallel.

Based on these properties, consider *gravity*. According to general relativity, in the absence of matter, space-time is flat. Then, when matter is introduced, this causes space-time to have curvature. This effect is given by the Einstein field equations $G = 8\pi T$, where T is the stress-energy tensor and G is the Einstein curvature tensor. It is for this reason that one can understand why the effect of gravity is manifested in the relationship between two moving particles; i.e., gravity causes geodesic deviation.

Let us now see what the New Foundations to Geometry say about this: Consider the *recoverability* of gravity in general relativity. Given a curved space-time manifold as the current data set, the physicist's inference, that gravity is responsible for the curvature, is an example of *external* inference, because the flat space-time is not within the current data set. Furthermore, according to the New Foundations to Geometry, flat space-time corresponds to an *iso-regular group,* in which a geodesic is a fiber, and the action of translating a fiber, parallel to itself, corresponds to a control group.

Therefore general relativity accords with the Externalization Principle of the New Foundations to Geometry. That is:

> **The inference, in general relativity, that geodesic deviation in curved space-time, leads back, in the absence of gravity, to the non-geodesic deviation of flat space-time, is an example of the Externalization Principle of the New Foundations to Geometry, because the inference is external and leads back to an iso-regular group.**

This also gives an explanation of special relativity: Since special relativity is the physics of flat space-time, special relativity corresponds to the iso-regular group recovered by fully externalizing general relativity. In fact, this is a deep explanation of special relativity because the explanation is an entirely general one that the New Foundations to Geometry give for *all* disciplines. Most crucially:

> **The New Foundations show that special relativity arises from the need to maximize recoverability.**

16 Quantum Mechanics

We shall now see that the Externalization Principle of the New Foundations to Geometry is fundamental also to quantum mechanics.

In quantum mechanics, any state $|\psi\rangle$ is a complex function, and the space of states is a (physical) Hilbert space of such functions. Given two states $|\psi\rangle$ and $|\phi\rangle$, their inner product is defined in this way:

$$\langle\psi|\phi\rangle \;=\; \int_a^b \psi^* \phi. \tag{27}$$

The associated norm is obviously given by $||\phi||^2 = \langle\phi|\phi\rangle$, which is related to the probability of the state.

An observable is a differential operator on this Hilbert space, and it induces a 1-parameter group on the space. In fact, one should think of any observable as belonging to a Lie algebra of observables, and its associated 1-parameter group is created by the usual exponentiation that goes from a Lie algebra to a Lie group. In quantum mechanics, one standardly considers the Lie algebra to be a collection of Hermitian operators, and the associated Lie group to be unitary. Thus the Lie group preserves the probability metric defined at (27) above, and is therefore an *isometry*, in fact, a rotation. Given an observable V, its associated 1-parameter group will be denoted by G_V.

Now, measurement with respect to an observable V does not destroy the information produced by another observable W only if the two observables commute, that is, if $[V, W] = 0$ within the Lie algebra of observables.

Now let us understand how the New Foundations describe this: According to the New Foundations to Geometry, any commuting pair of observables V and W, in quantum mechanics, should be corresponded to a wreath product of their 1-parameter groups G_W and G_V

$$[V, W] = 0 \qquad \longleftrightarrow \qquad G_W \,ⓦ\, G_V. \tag{28}$$

This is because G_V transfers the flow-lines of G_W onto each other, i.e., describes the flow-lines as <u>reused</u>.

Furthermore, since both G_W and G_V are isometry groups, the wreath product $G_W ⓦ G_V$ is *iso-regular,* i.e., it satisfies the three conditions of an iso-regular group. Therefore, we have this conclusion:

> **According to the New Foundations to Geometry, in quantum mechanics, two observables commute only if their 1-parameter groups form an iso-regular group.**

Now let us consider how physical structure is generated in quantum mechanics. One starts with a symmetric structure, e.g., an atom with a spherically symmetric Hamiltonian, and one successively adds asymmetries, in accord with the Asymmetry Principle of the New Foundations to Geometry. The initial symmetry corresponds to the commutation of observables, and successive addition of asymmetry corresponds to the successive breaking of the commutation. This means that the state is no longer described by the iso-regular group corresponding to the starting commutation.

For example, suppose one begins with a spherically symmetric Hamiltonian H. This means, in particular, that H and J_z commute, where J_z is the generator of rotations around the z-axis (i.e., the angular momentum observable for the z-axis). According to (28) above, this commutation corresponds to the wreath product $G_H ⓦ G_{J_z}$, which is *iso-regular*. The addition of an external asymmetrizing field will change H and can therefore destroy the commutation $[H, J_z]$. Thus, the asymmetrized state will not be described by the iso-regular group $G_H ⓦ G_{J_z}$.

Now consider the reverse-generation direction, i.e., the *inference* direction.

> **The above argument shows that, in quantum mechanics, a past symmetric state corresponds to an iso-regular group. This demonstrates that the Externalization Principle of the New Foundations to Geometry is fundamental to quantum mechanics.**

17 Hierarchical Reuse in Coordinate Systems

Coordinate systems are of course fundamental to all science data systems as well as all engineering design and manufacturing systems. Furthermore, all these systems use differential geometry, and certain coordinate systems are enormously important in revealing the crucial structure of differential manifolds.

The New Foundations to Geometry demonstrate that these important coordinate systems are all examples of the Externalization Principle; i.e., they are all derived from iso-regular groups. This section will illustrate this with one of the most powerful coordinate systems: *geodesic polar coordinates*. In doing so, we will demonstrate that geodesic polar coordinates are a hierarchical structure of *reuse*.

First, in this paragraph, we will recall the standardly stated properties of geodesic polar coordinates. These coordinates use the fact that, given a point p on a regular surface M, there is, within the tangent plane $T_p(M)$ to the surface at p, a neighborhood B of the origin of the tangent plane, such that the tangent vectors v in the neighborhood, possess the following properties, which will be illustrated using Fig 17. Note, of course, that a vector v can be written as tu, where u is the unit tangent vector in the direction of v, and t is a scaling of u. Now use the fact that, given any unit tangent vector u based at p, there is a unique surface geodesic γ, through p, that is parameterized by arc-length, and is in the direction of u, and therefore in the direction of v. Crucially, as illustrated in Fig 17, there is a well-defined map, called the *exponential map at* p, denoted by exp_p, that sends the vector $v = tu$ to the point $exp_p(v)$ on the geodesic γ, where this point $exp_p(v)$ is at arc-length t along the geodesic from the starting point p. That is, intuitively, one can think of the vector v as a straight line of length t and view the exponential map exp_p as wrapping this straight line along the geodesic γ on the surface so that the straight line of length t becomes a segment of length t along the geodesic. In fact, within $T_p(M)$, there is always a neighborhood B of the origin, such that exp_p restricted to that neighborhood is a *diffeomorphism* from B to a neighborhood A of p in the surface. The neighborhood A is called a *normal neighborhood* of p.

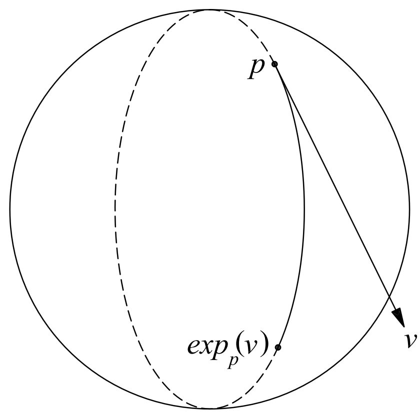

Figure 17: An example of the exponential map exp_p applied to a vector v tangential to a surface at p

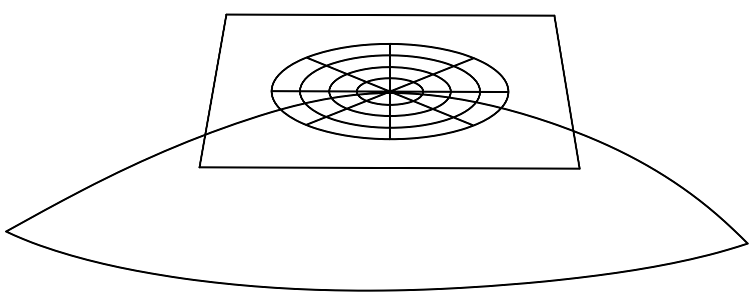

Figure 18: Within the tangent plane, T_pM, polar coordinates around the origin will be sent by the exponential map to geodesic polar coordinates on the surface. In particular, the radial lines of the polar coordinates of the tangent plane will be sent to the surface geodesics that are tangent to those radial lines.

Because the exponential map exp_p is a diffeomorphism from the neighborhood B to the normal neighborhood A, the map exp_p can provide coordinates for A. The most useful coordinates come from the polar coordinates (r, θ) of the tangent plane, which are illustrated in Fig 18. These are centered around the origin of the tangent plane. They are mapped by exp_p diffeomorphically from the neighborhood B in $T_p(M)$ to the neighborhood A in the surface. The resulting coordinates on A are called the *geodesic polar coordinates*. In the map exp_p, the circles of the polar coordinates in B are sent to curves in A that are called the *geodesic circles* in A. Observe that, based on what was said above, it is clear that the radial lines of the polar coordinates in $T_p(M)$ are mapped by exp_p to geodesics in the surface.[1] Notice that, along such a radial geodesic in the surface, the radial parameter r, in the polar coordinate (r, θ), gives the arc length along the geodesic. An important fact is that the geodesic circles are orthogonal to the radial geodesics.

Now let us look at the way the New Foundations to Geometry describe geodesic polar coordinates.

First, according to the New Foundations, the polar coordinates on a tangent plane T_pM are structured by an iso-regular group $\mathbb{R}$ ⓦ $SO(2)$. In this group, each fiber-group copy is the group of translations along a radial line through the origin of the tangent plane, and the control group $SO(2)$ transfers the fiber-group copies onto each other. This iso-regular group will be called the *planar radial iso-regular group* $\mathbb{R}$ ⓦ $SO(2)$ and is illustrated in Fig 19.

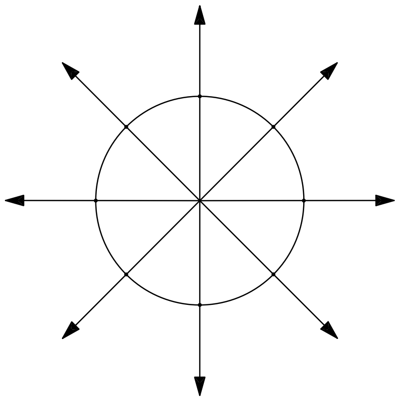

Figure 19: The planar radial iso-regular group $\mathbb{R}$ ⓦ $SO(2)$

Now, it is crucial to understand that the inference of the polar coordinates on the tangent plane, and the inference of the exponential map, which maps these coordinates onto the surface, is *external* inference. That is, the tangent plane is not within the data set visible to the observer; i.e., not within the surface.

Based on this, we can see the powerful fact that, once again, external inference accords with the Externalization Principle. That is, the inferred coordinates on the inferred tangent plane are structured by an iso-regular group.

[1] Standardly, one omits the radial line along $\theta = 0$. However, this is easily handled in our system.

Notice also that the fact that the exponential map sends the polar coordinates onto the surface means that the inferred iso-regular group $\mathbb{R} \circledS SO(2)$, which is not visible, is *transferred* onto the surface. This accords with the New Foundations fundamental principles: Maximization of Transfer and Maximization of Recoverability.

Now observe that because, within the tangent plane, the iso-regular group $\mathbb{R} \circledS SO(2)$ is a structure of *transfer*, and this structure of transfer is itself *transferred* onto the normal neighborhood of the surface, the situation is defined as *transfer of transfer*.

The next crucial concept to understand is this: In the tangent plane, each radial fiber-copy can be classified as a geodesic, and in the New Foundations to Geometry, its structure as a geodesic is given by the iso-regular group $\mathbb{R} \circledS \mathbb{R}$, where the control group $\mathbb{R}$ corresponds to the parameter along the radial line, and the fiber-group copies correspond to the tangent lines along the radial line. Therefore, the group $\mathbb{R} \circledS \mathbb{R}$ defines the geodesic as a transfer structure. Notice, in this planar geodesic, that the trace of the control group is *coincident* with the trace of each of the fiber-group copies.

Based on this, the New Foundations state that this transfer structure $\mathbb{R} \circledS \mathbb{R}$ of a radial line in the tangent plane is itself transferred onto a geodesic in the surface. The latter, being a geodesic, is a uniform transfer of its tangent fibers by parallel transport. The crucial fact is that, for the surface geodesic, the tangent fibers are not coincident with the geodesic curve. Therefore, we see that, although the surface geodesic is described internally by transfer, it is described *externally* as the transfer of an iso-regular group, i.e., the transfer structure of the corresponding radial tangent line as a geodesic. This again accords with the Externalization Principle.

GEODESIC POLAR COORDINATES
AS
<u>REUSE</u> OF <u>REUSE</u> OF <u>REUSE</u>

The New Foundations to Geometry represent geodesic polar coordinates in the following way:

A radial tangent line at a surface point is represented geodesically as an iso-regular group $\mathbb{R} \circledS \mathbb{R}$. This is transferred onto the other radial tangent lines, through that point, by the rotation group $SO(2)$, thus creating a *transfer of transfer* **structure, in the tangent plane, given by the iso-regular group:**

$$\mathbb{R} \circledS \mathbb{R} \circledS SO(2) \tag{29}$$

Then, this transfer hierarchy is transferred by diffeomorphism onto the surface, thus representing the surface as *transfer of transfer of transfer*.

The crucial fact is that the iso-regular group in expression (29) is inferred from the surface by the Externalization Principle.

Therefore, the New Foundations to Geometry justify and represent geodesic polar coordinates in terms of the principles of Maximization of Transfer and Maximization of Recoverability.

This gives the surface a representation that maximizes <u>reuse</u> as well as optimizing its <u>archival</u> function.

18 Computer-Aided Design

We shall now see that the Externalization Principle of the New Foundations to Geometery is fundamental to computer-aided design. The New Foundations demonstrate that shape primitives of computer-aided design are given by iso-regular groups. These are listed in Table 1.

The top half of the table shows what the New Foundations call the Level-Continuous Primitives. In these, each level is continuous. Since there are only two connected 1-parameter Lie groups, $SO(2)$ and $\mathbb{R}$, and since we must accord with the Maximization of Transfer principle, these primitives are generated simply by taking all possible 2-level wreath products using $SO(2)$ and $\mathbb{R}$.

LEVEL-CONTINUOUS	
Plane	$\mathbb{R} \ⓦ \mathbb{R}$
Sphere	$SO(2) \ⓦ SO(2)$
Cross-Section Cylinder	$SO(2) \ⓦ \mathbb{R}$
Ruled Cylinder	$\mathbb{R} \ⓦ SO(2)$

LEVEL-DISCRETE	
Cube	$\mathbb{R} \ⓦ \mathbb{R} \ ⓦ \ \mathbb{Z}_2 \ⓦ \mathbb{Z}_3$
Cross-Section Block	$\mathbb{R} \ⓦ \mathbb{Z}_n \ ⓦ \ \mathbb{R}$
Ruled or Planar-Face Block	$\mathbb{R} \ⓦ \ \mathbb{R} \ ⓦ \ \mathbb{Z}_n$

Table 1: CAD primitives as iso-regular groups

As an example, consider what the table calls a cross-section cylinder. This was the cylinder as given earlier in Fig 7, that is, the sweeping of a circular cross-section along its axis. The group we gave for that is

$$SO(2) \circledS \mathbb{R}.$$

The table shows also an alternative generation of a cylinder, which it calls the ruled cylinder. This time, a straight line within the cylinder surface is transferred around the cross-section. The transfer again is modeled by a wreath product. This reverses the fiber-control roles of $SO(2)$ and $\mathbb{R}$ from the previous case. Therefore, for this transfer structure, the wreath-product is this:

$$\mathbb{R} \circledS SO(2).$$

The lower half of the table gives what we call the level-discrete primitives. These are primitives in which one level of transfer is a discrete group. The cross-section block and ruled block correspond to the two cylinder cases just discussed, where the continuous rotation group $SO(2)$ is replaced by the discrete rotation group $\mathbb{Z}_n$, and an extra fiber level $\mathbb{R}$ is wreath sub-appended below the rotation group to correspond to the side.

The remaining entry in the table is the cube, which is a 3D version of the transfer structure we gave for the square in expression (13) page 21.

Now a crucial fact is that, given a CAD model, its inferred starting state is given by a shape primitive. According to the New Foundations to Geometry, the shape primitives are given by iso-regular groups. Therefore, the New Foundations demonstrate that CAD is fundamentally based on the Externalization Principle.

19 Interoperability of the Externalization Principle

**INTEROPERABILITY
OF THE
EXTERNALIZATION PRINCIPLE**

The New Foundations to Geometry prove that the Externalization Principle is necessarily the basis of *all* disciplines because the principle maximizes transfer and recoverability, and therefore maximizes reuse and archiving. Therefore, a crucial consequence is this:

The domain-independent validity of the Externalization Principle makes it a powerful tool for the integration of the many heterogeneous representations that occur in product lifecycles and data lifecycles; e.g., across entire sets of space missions, science data systems, manufacturing supply chains, etc.

20 Complexity

Complexity of data is a major factor in large-scale scientific and engineering systems.

As stated in the book *A Generative Theory of Shape*, the fundamental purpose of the New Foundations to Geometry is to handle complexity. This is achieved by a class of mathematical groups, invented in the New Foundations, called *unfolding groups*. Using these groups, the New Foundations give an extensive algebraic account of complex configurations in many disciplines, including computer-aided design, geology, computer vision, biological morphology, etc.

We are now going to describe unfolding groups. Consider the main problem in providing a generative theory of complex data.

> **According to section 12, recoverability is possible only if the generative operations are symmetry-breaking.**
> **Notice that, for complex data, using the Standard Foundations to Geometry, this would mean that, as one proceeds forward in the generative sequence, the symmetry group of the structure would quickly reduce to nothing. Thus, in the Standard Foundations, there would be a complete loss of algebraic information.**
> **The New Foundations solve this problem as follows:**

The New Foundations use the entirely opposite theory of symmetry-breaking, as described in section 12. In this theory, the group describing the symmetric past state is actually increased in the symmetry-broken state. This is done by making it the fiber group of a wreath product in which the control group is the group of the asymmetrizing action. Thus, in using the wreath product, the group of the past state is *transferred* onto the symmetry-broken state.

In previous sections, we have seen how this theory of symmetry-breaking gives a formulation of *deformation* in such a way that does not have the problems of the Standard Foundations to Geometry. For example, we saw, in the case of deforming a square into a parallelogram, and deforming a straight cylinder into a bent one, that the inferred symmetric past state is given by an iso-regular group, and the deformation is modeled by extending the iso-regular group, as a fiber group, by the deforming group, as a control group, in a wreath product; i.e., a structure of *transfer*.

Having shown how this theory of symmetry-breaking solves the problems of the Standard Foundations in modeling *deformation*, it is now necessary to show how the New Foundations solve the problems of modeling *concatenation*. For example, consider the intersection of two objects such as a cube and a cylinder. Each of the two objects *individually* has a high-degree of symmetry. However, their intersection looses much of this symmetry. Thus, the Standard Foundations would encode the concatenated structure by a reduced group. In contrast, the New Foundations to Geometry invented the opposite kind of group theory. In this, the group of the concatenated structure not only preserves the symmetry groups of the individual objects, but adds the extra information of the concatenation. This is done by *unfolding groups*, a class of groups invented by the New Foundations:

UNFOLDING GROUPS

Unfolding groups are characterized by the following two properties:

> **The control group acts *selectively* on only part of its fiber.**

> **The control group is symmetry-breaking by *misalignment*.**

Major classes of unfolding groups are structured by starting with a configuration in which n primitives are maximally aligned. This configuration will be called the *alignment kernel.* The unfolding causes successive misalignment of the primitives. Because this works by transfer, the unfolding action maps the alignment kernel onto misaligned versions of itself. Thus, in unfolding groups, the misaligned versions are mathematically described as the <u>reuse</u> of the original aligned state.

This is formalized in the following way:

(1) There are n objects which have symmetry groups $G_1, \ldots, G_n$. These correspond to the *primitives*. They are given by iso-regular groups.

(2) One forms the direct product, $G_1 \times \ldots \times G_n$, and makes this the fiber group of a wreath product, with control group $G(C)$, thus:

$$[G_1 \times \ldots \times G_n] \circledw G(C).$$

The direct product $G_1 \times \ldots \times G_n$ should not be confused with the product of fiber-group copies. It is a single fiber group.

(3) In the above wreath product, any fiber-group copy, i.e., any copy of $G_1 \times \ldots \times G_n$, corresponds to an *object configuration*.

(4) Let the fiber-group copy in which the object symmetry groups $G_1, \ldots, G_n$ are maximally aligned with each other, be called the *alignment kernel.* Choose this to be the fiber-group copy corresponding to the identity element of the control group.

(5) The control group transfers object-configurations $[G_1 \times \ldots \times G_n]_g$ onto object-configurations $[G_1 \times \ldots \times G_n]_h$. In doing so, it pulls the objects out of alignment with each other. The control group is therefore symmetry-breaking on the alignment kernel, by creating misalignment.

REPRESENTING COMPLEXITY

Unfolding groups were invented in the New Foundations to Geometry because these groups represent complex data in an understandable form by maximizing transfer and recoverability.

To begin to understand the applicability of unfolding groups, it is worth considering what the research procedure was, in the New Foundations, to create the group theory for complex structure in CAD. I worked through every single operation in each of several main CAD, solid modeling, assembly, and animation programs, including ProEngineer, AutoCAD, Architectural Desktop, Mechanical Desktop, 3D Studio Max, etc., as well as all the major manuals on each of the programs - approximately 15,000 pages of text. Each individual situation was characterized by a group, and a new class of groups was invented for any situation that could not be formalized in terms of any previously created class of groups. Proceeding in this manner, it was eventually found that three classes of groups could handle any newly created situation. I named them:

(1) Telescope groups.
(2) Super-local unfoldings.
(3) Sub-local unfoldings.

These are the main classes of unfolding groups. They will now be described.

21 Telescope Groups

Begin by considering Fig 20 which shows a concatenation of a cube and a cylinder. To represent this structure algebraically, the New Foundations to Geometry proceed as follows: The generative history starts with the two independent objects, and therefore the symmetry of this starting situation is given thus:

$$G_{cylinder} \times G_{cube}$$

which is the *direct product* of the *iso-regular groups*, $G_{cylinder}$ and G_{cube}, that describe the two independent objects.

Now, by the maximization of transfer, the starting group, i.e., this direct product group, must be transferred onto subsequent states in the generative history, and therefore it must be the fiber of the wreath product in which the control group creates the subsequent generative process.

Let us take the control group to be the *affine group* $AGL(3, \mathbb{R})$ on three-dimensional real space.The full structure, fiber plus control, is therefore the following wreath product:

$$[G_{cylinder} \times G_{cube}] \circledS AGL(3, \mathbb{R}).$$

Now, it is necessary to fix the *group representation* of this wreath product. First, by our theory of recoverability, the control group must have an asymmetrizing action. Thus proceed as follows: The particular fiber-group copy

$$[G_{cylinder} \times G_{cube}]_e$$

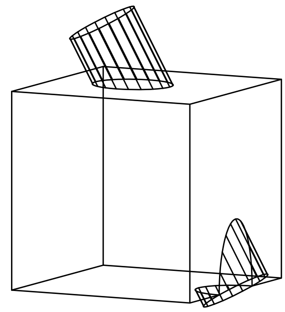

Figure 20: Concatenation of cylinder and cube.

corresponding to the identity element e in the affine control group, must be the most symmetrical configuration possible. This exists only when the cube and the cylinder are positioned with their symmetry structures (axes, etc.) *maximally aligned*. This is the configuration that I call the **alignment kernel**.

Next, choose one of the two objects to be a reference object. This will remain fixed at the origin of the world coordinate frame. Let us choose the cube as the referent. Given this, now describe the action of the affine control group as providing an affine motion of the cylinder *relative* to the cube. Each fiber-group copy

$$[G_{cylinder} \times G_{cube}]_g$$

for some member g, of the control affine group, is therefore an arrangement of this system. In fact, any fiber copy will be called a **configuration** of the system. For example, Fig 20 corresponds to a configuration. The crucial concept is this:

<u>REUSE</u> OF CONFIGURATIONS

> **The role of the control group is to transfer configurations onto configurations; i.e., to <u>reuse</u> configurations**.

The wreath product we have presented so far:

$$[G_{cylinder} \times G_{cube}] \circledR AGL(3, \mathbb{R})$$

gives the *complete* symmetry group of the concatenated situation. It has all the internal symmetries of the objects individually, as well as their relationships.

Let us now understand how to add a further object, for example a sphere. First of all, the fiber becomes the following, with the added sphere group:

$$G_{sphere} \times G_{cylinder} \times G_{cube}.$$

Let us define the cube as the referent for the cylinder-sphere pair, and the cylinder as the referent for the sphere.

Accordingly, there are now two levels of control, each of which is the affine group $AGL(3,\mathbb{R})$, and each of which is added via a wreath product. Thus we obtain the 3-level wreath product:

$$[G_{sphere} \times G_{cylinder} \times G_{cube}] \circledS AGL(3,\mathbb{R}) \circledS AGL(3,\mathbb{R}).$$

This is interpreted in the following way: Initially, the three objects (sphere, cylinder, cube) are coincident with their symmetry structures maximally aligned. This corresponds to the fiber-group copy that we call the *alignment kernel*. The higher affine group moves the sphere-cylinder pair in relation to the cube. The lower affine group moves the sphere in relation to the cylinder.

Recall that, in the above situation, the cube is fixed at the origin of the world-frame. In fact, in our theory, its symmetries are maximally aligned with the symmetries of the world frame. Now, if we also allow the cube to move with respect to the world-frame, then we add the group G_W, defining the symmetries of world frame, into the alignment kernel, and add a third level of control above the two control levels that have already been included, thus:

$$[G_{sphere} \times G_{cylinder} \times G_{cube} \times G_W] \circledS AGL(3,\mathbb{R}) \circledS AGL(3,\mathbb{R}) \circledS AGL(3,\mathbb{R}).$$

The new top control level will move the cube with respect to the world-frame. Notice that the full control group corresponds to that in the group of the entire transform structure given section 8 in the Mathematical Theory of Object-Linked Inheritance. In my book *A Generative Theory of Shape*, there is a chapter devoted to giving an algebraic theory of reference frames, and I show that the appropriate group G_W, for the world frame, is the iso-regular group that is the maximal normal subgroup of the hyperoctahedral group.

The crucial point here is that, initially, the four objects (sphere, cylinder, cube, world-frame) are coincident with their symmetry structures maximally aligned. This corresponds to the fiber-group copy that I call the *alignment kernel*. The hierarchy of control groups move these objects hierarchically out of alignment with each other, in correspondence with the inheritance hierarchy. This is an example of a *telescope group*, one of the classes of groups invented in the New Foundations to Geometry.

Another crucial fact is that the above discussion illustrated the theory of *feature attachment* given by the New Foundations. Feature attachment is the term used in mechanical design for the successive addition of structural units and components. Clearly, it is a very important phenomenon in the creation of complex structures.

MATHEMATICAL THEORY
OF
FEATURE ATTACHMENT

The New Foundations to Geometry give the following mathematical theory of feature attachment: When one creates objects and *attaches* them in the generative structure, one is entering new instances into the alignment kernel, and positioning the command group for each new instance in the appropriate wreath position within the unfolding group corresponding to the inheritance hierarchy of the structure.

22 Super-Local Unfoldings

Super-local unfolding groups are another class of unfolding groups invented in the New Foundations. To illustrate them, consider the following frequent situation in design: The designer selects part of the existing design, copies it, and then drags the copy to some other region of the design, perhaps with modification; e.g., walls are created not by drawing a new wall each time but by copying, moving, and modifying existing walls. The New Foundations model this by an unfolding group structured in the following way:

$$[G_1 \ldots G_j] \circledw G_n^X .$$

The group in brackets represents a shape, e.g., a design, up to the current state. Then, one wreath-appends a group G_n above this, which acts selectively on only some part X of the structure below. The wreath product operation here indicates that G_n acts by *transferring* X in some way. The entire group is called *super-local* because it is created by wreath-appending a control group *above* an existing structure, such that the added control group acts selectively on only part of its fiber. Such groups model situations, for example, in *AutoCAD*, where one freezes part of the existing structure and manipulates some unfrozen cross-hierarchy selection of elements; or conversely, situations, for example, in *3D Studio Max*, where the cross-hierarchy selection is locked and manipulated over a sequence of steps.

23 Sub-Local Unfoldings

Sub-local unfolding groups are another class of unfolding groups invented in the New Foundations.

> **The power of sub-local unfoldings is that they handle anomalies.**
>
> **According to the New Foundations:**
>
> **Complex shape generation is the generation of anomalies.**
>
> **They are the essense for example of crystal physics, i.e., you can only really see a crystal via its defects.**
>
> **Any environment, e.g., a complex scene in computer vision, is, according to this theory, a hierarchy of anomalies.**
>
> **It is in order to handle this, that the New Foundations invented sub-local unfolding groups.**

To illustrate sub-local unfoldings, consider Fig 21. It could represent many situations, e.g., in crystal physics, mechanical design, etc. In this structure, there is a large central object, the highest parent level in the inheritance hierarchy, and six surrounding child components. Furthermore, on the 5th component of this child-level, there is added an extra child with respect to that component, as shown in the bottom right of the figure.

Because the 5th component is the only one that has this extra child object, this extra object is an *anomaly*.

Let us first give the group of the structure *without* the anomalous extension, thus:

$$\begin{aligned}&[\mathbb{R} \,\circledS\, \mathbb{Z}_4]_{\mathcal{U}}\\&\circledS[AGL(2,\mathbb{R}) \times AGL(2,\mathbb{R}) \times AGL(2,\mathbb{R}) \times AGL(2,\mathbb{R}) \times AGL(2,\mathbb{R}) \times AGL(2,\mathbb{R})]\\&\circledS AGL(2,\mathbb{R}).\end{aligned} \tag{30}$$

The affine groups $AGL(2,\mathbb{R})$ are the *command* groups associated with the object instances. They are arranged in accord with the algebraic theory of inheritance; that is, a parent-child relationship is given by a wreath product, and a parallel relationship is given by a direct product. Thus to interpret this group: The affine group on the bottom line gives the relation between the central object and the world frame. The six affine groups on the middle line give the relation between the six surrounding objects and the central object. The top line represents the alignment kernel, which consists of a direct product of as many instances of the primitive $\mathbb{R} \,\circledS\, \mathbb{Z}_4$, as are required.

Now observe that the wreath product symbol at the beginning of the bottom line says that the relation between the central object corresponding to the bottom line, and

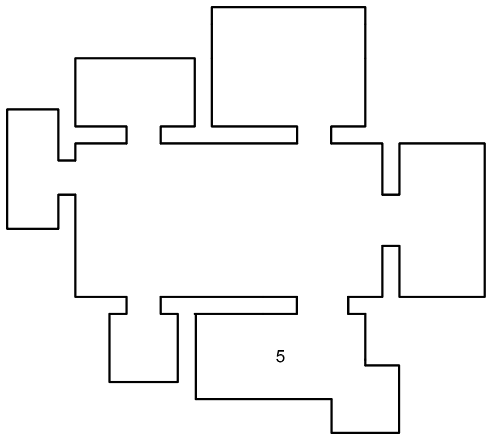

Figure 21: The addition of an anomaly to the 5th component in 6-fold child-object level.

the six objects corresponding to the middle line, is that of parent to children. More properly, this is a relation between the *command groups* of those object instances. The direct products in the middle line says that the relation between the command groups of the six surrounding objects is parallel.

Now, to add the *anomaly*, i.e., the extension object off object 5 in the child level, we wreath sub-append an extra affine group to the 5th member of the direct product in the second line, and add an extra iso-regular group into the alignment kernel. This illustrates a *sub-local unfolding group*, one of the classes of groups invented in the New Foundations to Geometry.

It also illustrates the Theory of Feature Attachment stated in section 21.

The extensive algebraic theory developed for this in the book *A Generative Theory of Shape* is applied at length to mechanical design and manufacturing, architectural design, and robotics (as well as perception). Using the theory of unfolding groups, the book works in detail through the main stages of mechanical CAD/CAM: part-design, assembly and machining. For example, in part-design, the book gives an extensive algebraic analysis of sketching, alignment, dimensioning, resolution, editing, sweeping, feature-addition, and intent-management. The equivalent analysis is also done for architectural design. The structure of robot manipulators and assembly is also a central concern.

The next section will illustrate how I have applied the theory of unfolding groups to crucial structures in earth sciences and biological morphology.

24 Process-Grammar for Science Data Systems and Engineering

One part of the New Foundations for Geometry is called the *Process-Grammar*. It gives a mathematical theory in which shape bifurcation is actually shown to be an example of reuse.

I invented the Process-Grammar in the 1980s as a mathematical theory that defines and models crucial aspects of *morphology*. Since its publication, scientists have applied the Process-Grammar and parts of its mathematics in many disciplines such as earth sciences, including volcanic island formation, meteorology, and drainage patterns; as well as numerous areas of biology and medicine, including cardiac diagnosis, MRI human brain scans, dental radiographs, transmission electron microscope (TEM), musculoskeletal development, the imaging of neurons, the analysis of abnormal anatomy with applications in radiotherapy, surgery, and psychiatry, the tracking of DNA molecules, radiology, the morphology of fish, the morphology of leaves; it was also applied in computer-aided design. References include Milios [25]; Lin, Liang & Chen [19]; Larsen [9]; Mayoh [24]; Deguchi & Furukawa [6]; Pernot, Guillet, Leon, Falcidieno, & Giannini [28]; Shemlon [33]; De Sa, Radice & Kerckhove [5]; Costa [4]; Pizer, Fritsch, Yushkevich, Johnson, & Chaney [29]; Parvin, Peng, Johnston & Maestre [27]; Ogniewicz [26]; Lopez [21]; Torres & Falcão [34].

A basic aspect of the Process-Grammar is a new definition of symmetry that has properties fundamentally different from standard definitions of symmetry. Furthermore, we will see that its fundamentally different properties capture crucial aspects of scientific and engineering data, which the standard definitions fail to do.

This Process-Grammar definition of symmetry is illustrated in Fig 22. Consider two curves c_1 and c_2 as shown in the figure. A symmetry point between the two curves is defined as follows: Place a circle such that it is simultaneously tangential to the two curves. The symmetry point is the point shown as Q, which is the mid-point on the shorter circle arc between the two tangent points. As the circle moves between the two curves, maintaining the double-tangency property (thus having to contract and expand), the trace of the symmetry points Q is defined to be the symmetry axis. I call this axis the *Process-Inferring Symmetry Axis*, PISA, because, as we will see, it infers crucial processes.

It is important to note that the conventional symmetry axis that uses a double-tangency property is the Medial Axis of Blum [2]. This defines the symmetry axis as the trace of circle centers. In my book *Symmetry, Causality, Mind* (MIT Press), I showed that the Medial Axis produces symmetry axes that are topologically fundamentally different from the PISA axis. Furthermore, I have shown that the Medial Axis completely fails to describe morphology. In contrast, PISA captures crucial aspects of morphology, as we shall see.

I will now state a theorem that I proved in the 1980s, which has been used in an enormous number of disciplines:

SYMMETRY-CURVATURE DUALITY THEOREM
Leyton (1987)

Given a smooth curve with one and only one curvature extremum, it has one and only one PISA axis. Also it has one and only one Medial Axis.

The PISA axis always terminates at the curvature extremum. The Medial Axis always fails to terminate at the curvature extremum.

For this reason, the PISA axis is appropriate for describing morphology, and the Medial Axis is completely inappropriate for describing morphology.

Now a fundamental reason for the appropriateness of the PISA axis is the following rule in the Process-Grammar:

INTERACTION PRINCIPLE (Leyton, 1984): *PISA symmetry axes are the directions along which processes are hypothesized as most likely to have acted.*

According to the Process-Grammar, the above two rules, the Symmetry-Curvature Duality Theorem and the Interaction Principle, together infer crucially important aspects of the history of a shape. That is, the two rules imply that the symmetry axes leading to the extrema are the trajectories along which the extrema traveled as they were being created.

To obtain extensive corroboration for the two rules, we now apply them to a large catalogue of shapes: all shapes with up to, and including, eight curvature extrema. The results are shown in Fig 23, 24, and 25.

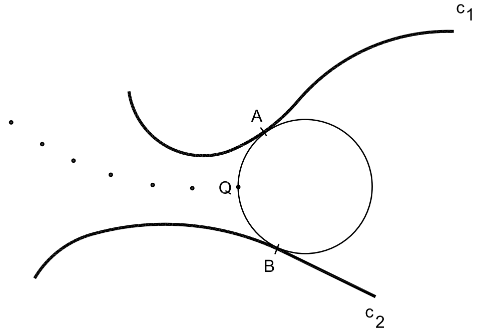

Figure 22: In the PISA system, the points Q define the symmetry axis.

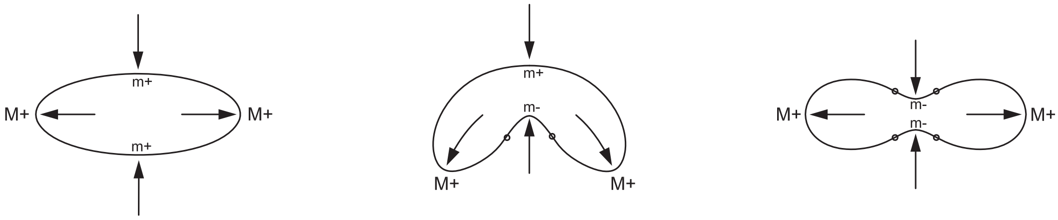

Figure 23: The inferred histories on the shapes with 4 extrema.

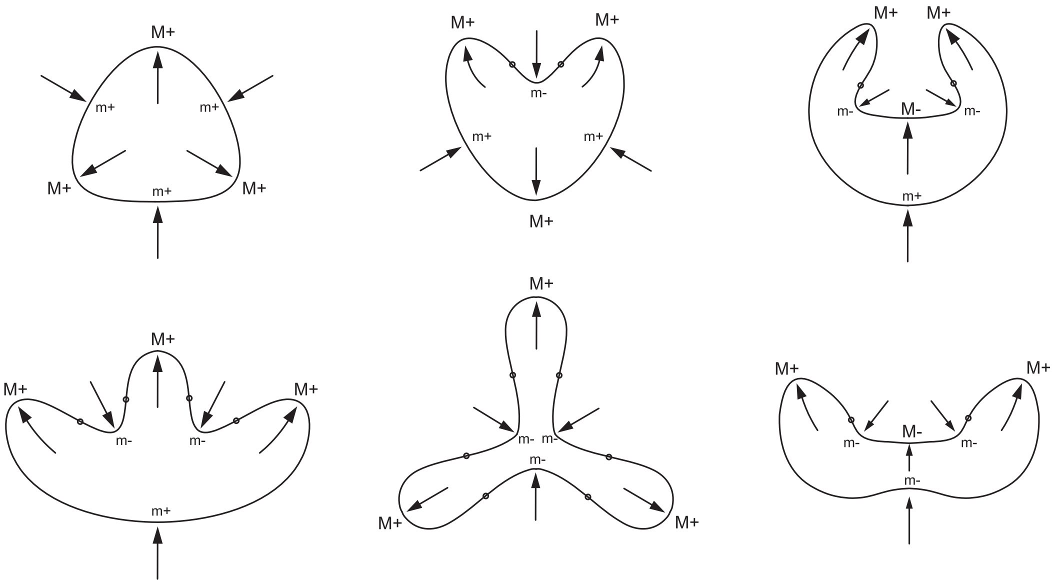

Figure 24: The inferred histories on the shapes with 6 extrema.

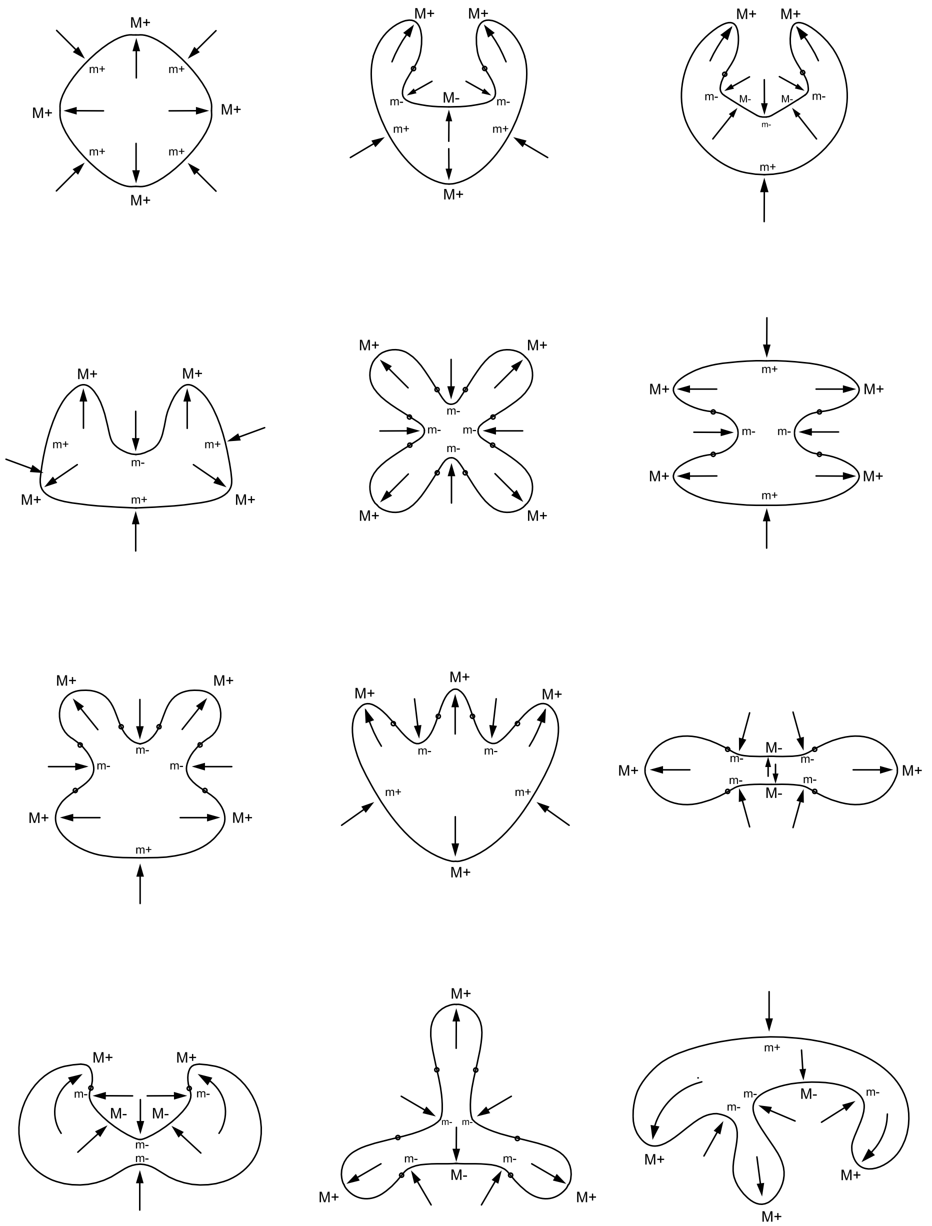

Figure 25: The inferred histories on the shapes with 8 extrema.

Most of the *outlines* in Fig 23, 24, and 25. came from a paper by Richards, Koenderink & Hoffman [32], and the Process-Grammar was used to complete the catalogue.

What I did was apply the Symmetry-Curvature Duality Theorem and the Interaction Principle to these outlines. These two rules produced the arrows on each shape, indicating how the shapes were formed over time. As the reader can see, these inferred histories accord very strongly with one's sense of how these shapes were formed.

Now, on every shape in Fig 23 - 25, each curvature extremum is labeled by one of four symbols M^+, m^-, m^+, M^-, which classifies the extremum. The classification is given as follows: First, define the curvature function along a curve as follows: Consider the curve as the *boundary* of an object, and travel along the curve in the direction that keeps the boundary on the *left* side of the curve. Then define curvature as the rate of *anti-clockwise* rotation. Fig 26 illustrates the curvature function along the parameter of a curve. There are four types of extrema, as shown by the four labels on the graph.

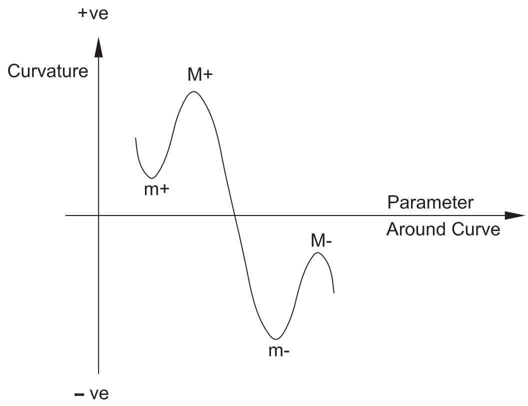

Figure 26: A curvature function showing the four types of extrema.

It is important to understand the shapes of curves corresponding to the four types of extrema. They are illustrated in Fig 27. In each case, the curve is defined as a boundary of an object, which is given by shading.

The arrows shown in Fig 27 are the PISA axes of the four curves. In accord with the Symmetry-Curvature Duality Theorem, they each terminate at the extremum on their curve.

The crucial fact is this: In the entire 2500 year history of symmetry, the symmetry axis of each of the four curves shown in Fig 27 would be on the *convex* side of the curve, i.e., below each of these curves. In contrast, the PISA axis is on the convex side of the first two curves, but on the concave side of the other two curves. We are going to see that this violation of the conventional view of symmetry is fundamentally important for understanding important structures in earth sciences, biology, engineering design, etc.

Now notice the following important phenomenon: In surveying the curves in Fig 27 it becomes clear that the four extrema types correspond to four English terms that people use to describe *processes*. Table 2 gives the correspondence.

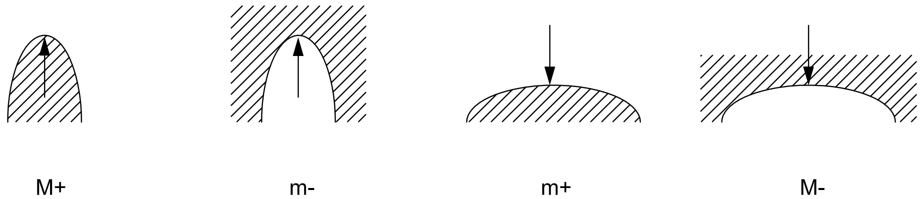

Figure 27: The four types of extrema on curves, and their PISA axes.

Notice that the conventional symmetry axes would not be able to give these terms. For example, with the third extremum, all the conventional symmetry axes would be *below* the third curve in Fig 27, and thus would not correctly represent squashing. It is only the PISA axis that correctly represents the squashing process that created this extremum.

EXTREMUM TYPE	$\longleftrightarrow$	**PROCESS TYPE**
M^+	$\longleftrightarrow$	protrusion
m^-	$\longleftrightarrow$	indentation
m^+	$\longleftrightarrow$	squashing
M^-	$\longleftrightarrow$	internal resistance

Table 2: Correspondence between extremum type and process type.

In accord with the fact that the PISA process arrows for the first two extrema correctly explain the *sharpening* of those extrema, and the PISA process arrows for the other two extrema correctly explain the *flattening* of those extrema, the Process-Grammar calls the first two extrema, **penetrative extrema**, and the other two extrema **compressive extrema**.

A crucial aspect of the Process-Grammar is that it defines the operations by which the processes can undergo bifurcation. We will now describe the four bifurcations that occur in level-3 of the Process-Grammar. They are as follows:

SHIELD FORMATION

This is illustrated in Fig 28. At the top of the first shape, there is a curvature extremum which is a positive maximum, M^+. Under the inference rules of the Process-Grammar, the force that created this extremum is given by the upward arrow leading to this extremum. In the transition to the second shape, this M^+ extremum and its force undergo a bifurcation into two copies, given by the left-ward and right-ward pointing arrows in the second shape. Necessarily, at the top of the second shape, a new extremum is introduced, a positive minimum, m^+.

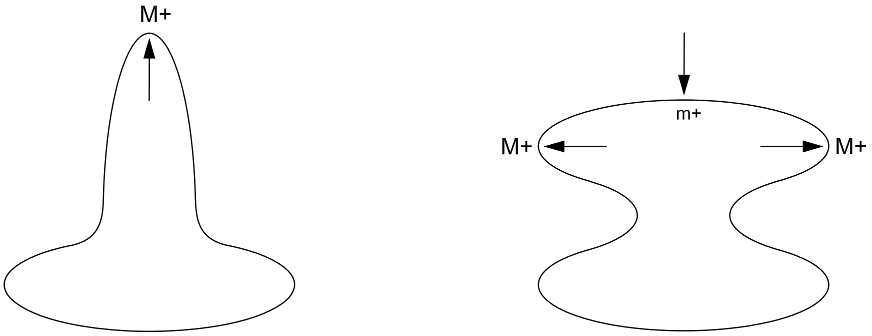

Figure 28: The shield-formation operation of the Process-Grammar.

Therefore, the transition from the first shape to the second shape is coded by the following operation from the Process-Grammar:

$$BM^+ \ : \ M^+ \ \longrightarrow \ M^+m^+M^+$$

That is, the M^+ extremum, at the top of the first shape, bifurcates into the triple $M^+m^+M^+$, at the top of the second shape.

Now, according to the Process-Grammar, the force that caused the transition in Fig 28 is given by the *downward* arrow leading to the m^+ extremum at the top of the second shape. That is, the transition was caused by this downward force pressing against the top upward force in the first shape, and causing that *outward* force to split to the left and right.

BAY FORMATION

This is illustrated in Fig 29. In the middle of the first shape, there is a curvature extremum which is a negative minimum, m^-. Under the inference rules of the Process-Grammar, the force that created this extremum is given by the downward arrow leading to this extremum. In the transition to the second shape, this m^- extremum and its force undergo a bifurcation into two copies, given by the left-ward and right-ward pointing arrows in the second shape. Necessarily, in the center of the second shape, a new extremum is introduced, a negative maximum, M^-.

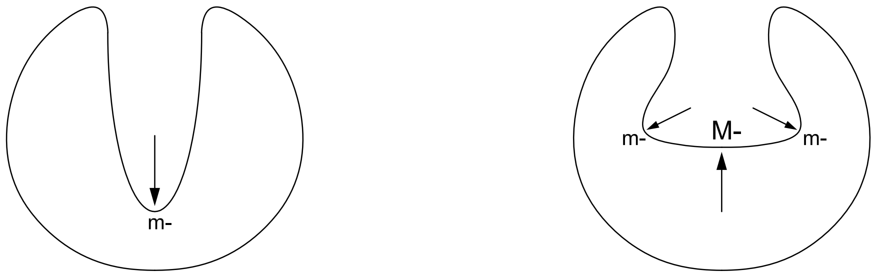

Figure 29: The bay-formation operation of the Process-Grammar.

Therefore, the transition from the first shape to the second shape is coded by the following operation from the Process-Grammar:

$$Bm^- \; : \; m^- \;\longrightarrow\; m^-M^-M^-$$

That is, the m^- extremum, shown in the first shape, bifurcates into the triple $m^-M^-m^-$, shown in the second shape.

Now, according to the Process-Grammar, the force that caused the transition in Fig 29 is given by the *upward* arrow leading to the M^- extremum in the center of the second shape. That is, the transition was caused by this upward force pressing against the downward force in the first shape, and causing that force to split to the left and right.

BREAKING-THROUGH OF A PROTRUSION

This is illustrated in Fig 30. At the top of the first shape, there is a compressive extremum which is a positive minimum, m^+. Under the inference rules of the Process-Grammar, the force that created this extremum is given by the downward arrow leading to this extremum. In the transition to the second shape, this m^+ extremum and its force undergo a bifurcation into two copies, given by the two outside diagonal arrows in the second shape. Necessarily, at the top of the second shape, a new extremum is introduced, a positive maximum, M^+.

Therefore, the transition from the first shape to the second shape is coded by the following operation from the Process-Grammar:

$$Bm^+ \; : \; m^+ \;\longrightarrow\; m^+M^+m^+$$

That is, the m^+ extremum, at the top of the first shape, bifurcates into the triple $m^+M^+m^+$, at the top of the second shape.

Now, according to the Process-Grammar, the force that caused the transition in Fig 30 is given by the *upward* arrow leading to the M^+ extremum at the top of the second shape. That is, the transition was caused by this upward force pressing against the top downward force in the first shape, and causing that downward force to split to the left and right. Since the upward force emerges as a protrusion in the second shape, this Process-Grammar operation is called *breaking through of a protrusion*.

Figure 30: The breaking-through of a protrusion operation of the Process-Grammar.

BREAKING-THROUGH OF AN INDENTATION

This is illustrated in Fig 31. In the center of the first shape, there is a compressive extremum which is a negative maximum, M^-. Under the inference rules of the Process-Grammar, the force that created this extremum is given by the upward arrow leading to this extremum. In the transition to the second shape, this M^- extremum and its force undergo a bifurcation into two copies, given by the left-ward and right-ward upward diagonal arrows in the second shape. Necessarily, in the middle of the second shape, a new extremum is introduced, a negative minimum, m^-.

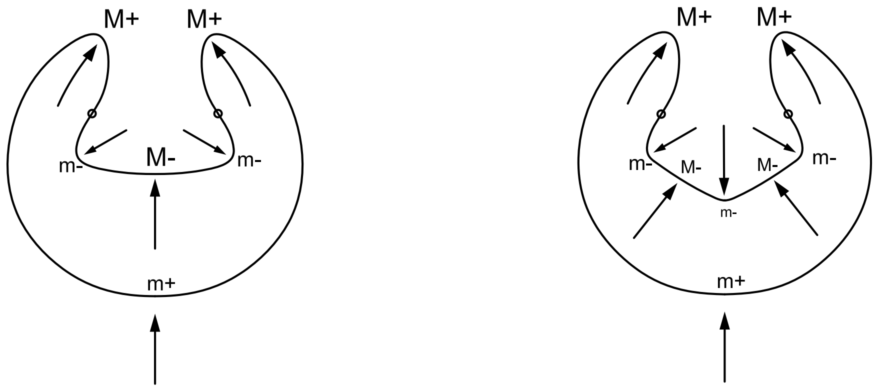

Figure 31: The breaking-through of a indentation operation of the Process-Grammar.

Therefore, the transition from the first shape to the second shape is coded by the following operation from the Process-Grammar:

$$BM^- \ : \ M^- \ \longrightarrow \ M^-m^-M^-$$

That is, the M^- extremum, in the center of the first first shape, bifurcates into the triple $M^-m^-M^-$, in the second shape.

Now, according to the Process-Grammar, the force that caused the transition in Fig 31 is given by the *downward* arrow leading to the m^- extremum in the center of the second shape. That is, the transition was caused by this downward force pressing against the central upward force in the first shape, and causing that upward force to split to the left and right. Since the downward causing force emerges as the indentation in the second shape, this Process-Grammar operation is called *breaking through of an indentation*.

25 Shape Bifurcation as <u>Reuse</u>

We will now show that:

> **According to the New Foundations to Geometry, the shape bifurcation operations are mathematically structured by <u>reuse</u>.**
>
> **This is given by unfolding groups (p 50), which the New Foundations invented to describe complexity in terms of <u>reuse</u>.**

As an example, consider the shape-bifurcation operation

$$Bm^+ : m^+ \longrightarrow m^+ M^+ m^+$$

which codes the situation, *breaking-through of a protrusion*, illustrated in Fig 30.

With respect to this, it is necessary to consider what this operation does to the *curvature function* of the shape. This is illustrated by the sequence of three functions in Fig 32. We see that the first function has the singularity-configuration m^+, which is the domain of the above operation Bm^+. And the third function has the singularity-configuration $m^+ M^+ m^+$, which is the codomain of the operation.

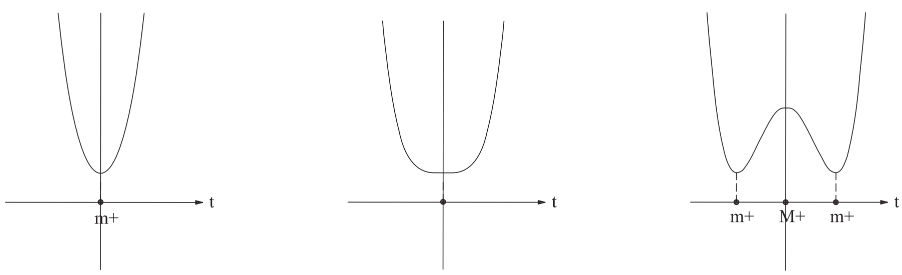

Figure 32: The succession of curvature functions in the Bm^+ bifurcation operation.

Now, the Process-Grammar uses the crucial assumption that each singularity in the domain and codomain is non-degenerate, i.e., not the *coincidence* of multiple singularities. However, since the Process-Grammar operation describes a bifurcation of the initial non-degenerate singularity into three non-degenerate singularities, the curvature function must have had a transition state in which the domain non-degenerate singularity m^+ became a 3-fold degenerate singularity, i.e., the *coincidence* of three singularities. This state is illustrated in the middle function in Fig 32.

Now consider the Symmetry-Curvature Duality Theorem. An important aspect of my *proof* of this theorem must be understood, as follows. Observe that there must be a neighborhood of the extremum in which the curve on each side of the extremum is a spiral (a spiral is a curve of monotonically increasing or monotonically decreasing curvature). Part of my proof of the theorem was a proof that a spiral cannot have a symmetry axis, Leyton [13]. Next, consider one of the circles in the trajectory of doubly-tangential circles leading to the extremum. The circle touches the curve at two points, A and B, on the two sides of the curvature extremum. Furthermore, it defines a reflection between the tangent-line at A and the tangent-line at B. Now, as the successive bitangent circles go towards the curvature extremum, the two tangent-lines, one on each side of the extremum, successively *converge*, and become coincident at the extremum.

Therefore, according to the New Foundations, the extremum is defined by the following iso-regular group

$$\mathbb{R} \circledwedge \mathbb{Z}_2 \tag{31}$$

which is explained as follows: In this wreath product, there are two copies of the fiber group, $\mathbb{R}$, corresponding to the two elements of the reflection control group $\mathbb{Z}_2$. These two copies of $\mathbb{R}$ correspond to the two tangent-lines, which have become coincident at the extremum. The control group $\mathbb{Z}_2$ *transfers* the two copies of $\mathbb{R}$, i.e., the two tangent-lines, onto each other.

Notice that this wreath product also describes the symmetry that relates the two tangent-lines for any doubly-tangential circle in the trajectory. It is only at the extremum, that the two tangent-lines become coincident with each other, and their PISA symmetry point Q becomes an actual point on the curve, i.e., the extremum. Thus, the extremum is the only curve-point that has the symmetry given by the above wreath product.

In relation to this, let us now apply the Mathematical Theory of Feature Attachment given on page 54. The first thing this theory says is this: In creating and attaching objects into the structure, one enters them into the alignment kernel.

Therefore, in the transition from the non-degenerate minimum to the 3-fold degenerate minimum, one is adding two new objects into the alignment kernel, as follows: (1) One is *cloning* the existing minimum, and (2) one is adding a maximum. Thus, the alignment kernel is given as follows:

$$(\mathbb{R} \circledwedge \mathbb{Z}_2)_{m^+} \times (\mathbb{R} \circledwedge \mathbb{Z}_2)_{m^+} \times (\mathbb{R} \circledwedge \mathbb{Z}_2)_{M^+} \tag{32}$$

where each of the three components is the iso-regular group $\mathbb{R} \circledwedge \mathbb{Z}_2$, given in expression (31) for an extremum, and each is labeled by the non-degenerate extremum that it will become *after* bifurcation.

Next, the Mathematical Theory of Feature Attachment says this: In entering new instances into the alignment kernel, one positions the command group for each new instance in the appropriate wreath position within the unfolding group corresponding to the inheritance hierarchy of the structure.

To understand the command groups added into the control group, first observe the following: A valuable way of describing the bifurcation is to define the two minima as diverging *relative* to the central maximum. This means that, in the inheritance hierarchy, *both* minima are children of the central maximum.

Thus, taking the alignment kernel in expression (32), and adding the control structure that defines the movement of the two minima, we obtain the following unfolding group:

$$\begin{aligned}&[(\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{M^+}] \\ &ⓦ \quad [\, AGL(2,\mathbb{R})_{m^+} \;\times\; AGL(2,\mathbb{R})_{m^+} \,].\end{aligned}$$

To help understand this structure, the group has been written on two lines. The first line is the fiber, which is the alignment kernel. The second line is the control group. This is seen from the fact that the wreath-product symbol is at the beginning of the second line. The important thing to observe is that the control group is the *direct product* of two affine groups $AGL(2,\mathbb{R})$, corresponding to the two m^+ extrema. These two affine groups will move the two m^+ extrema *relative* to the M^+ extremum. That is, they will move the two m^+ iso-regular groups, in the alignment kernel, relative to the M^+ iso-regular group, in the alignment kernel. The fact that the relation between the two affine groups is a direct product captures the fact that they are on the *same level* in the control group, which captures the fact that the two m^+ copies are on the same level in the inheritance hierarchy; i.e., they are both children of the M^+ extremum.

Thus, in the above unfolding group, the two copies of the affine group will *misalign* the *symmetries of the two minima* with respect to the *symmetries of the maximum*.

Now, the above structure defines the world-frame as fixed to, and aligned with, the maximum. There are some situations where this assumption can be useful; e.g., if the observer is traveling on the central extremum.

In other cases, where we wish to understand the maximum as moving relative to the world-frame, we expand the alignment kernel to contain the group G_W of the world-frame, thus:

$$(\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{M^+} \times G_W \tag{33}$$

According to the New Foundations, the group G_W of the world-frame is itself also a reflection structure, in fact, the reflection group given for the square in expression (13) page 21. Therefore one of its fiber reflection axes must initially be coincident with the reflection axis of the extremum.

By adding the world-frame as an extra object, we have expanded the *inheritance* hierarchy thus: Both minima remain children of the maximum, but the maximum has now become a child of the world-frame.

Thus, taking the alignment kernel in expression (33), and adding the control structure that corresponds to the inheritance hierarchy, we obtain the following unfolding group:

$$\begin{aligned}&[(\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{m^+} \times (\mathbb{R} ⓦ \mathbb{Z}_2)_{M^+} \times G_W] \\ &ⓦ \quad [\, AGL(2,\mathbb{R})_{m^+} \;\times\; AGL(2,\mathbb{R})_{m^+} \,] \\ &ⓦ \quad AGL(2,\mathbb{R})_{M^+}.\end{aligned}$$

This group is now a wreath product of three levels, which have been put on three lines, to help understand the structure. The first line is the fiber, which is the alignment kernel. The second line is level 1 of the control group, which, as previously, is the direct product of the two affine groups that misalign the two minima with respect to the maximum. The third line is level 2 of the control group, which has been added to misalign the maximum with respect to the world-frame.

Notice that both the second and third lines begin with a wreath-product symbol, which indicates that these are levels in the control group, and also indicates the *object-linked inheritance hierarchy*, which, according to the New Foundations, is coded by the wreath hierarchy.

This illustrates the fact that, according to the New Foundations to Geometry, the shape bifurcation operations are mathematically structured by reuse, and that this is given by unfolding groups, which the New Foundations invented to describe complexity in terms of reuse.

26 Object-Oriented Programming: Class Inheritance

The New Foundations to Geometry give New Foundations to Object-Oriented Programming, including inheritance, object-creation, class structure, class consistency, command structure, software text, etc. Most of the remainder of this paper will illustrate this. First, this section presents our theory of class inheritance. Let us begin by considering a typical class-inheritance hierarchy for closed figures, based on Meyer [23] p528. It is shown as Fig 33.

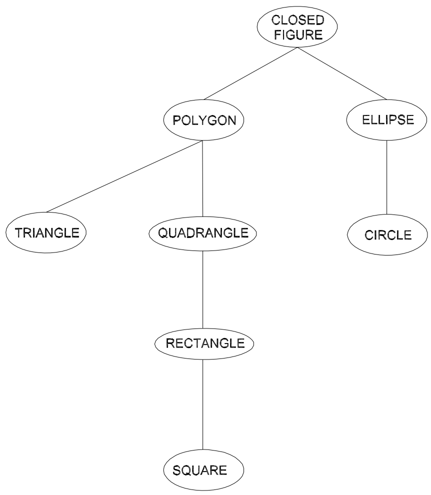

Figure 33: A typical class-inheritance hierarchy based on Meyer [23] p528.

Current object-oriented programming has no systematic way of explaining such a hierarchy. However, the New Foundations explain this with complete rigor and insight, as follows:

The two basic principles of the Foundations are (1) Maximization of Transfer, and (2) Maximization of Recoverability. Let us begin by using the second principle. This is realized by the Asymmetry Principle which recovers symmetries from asymmetries. In particular, the Externalization Principle says that any use of the Asymmetry Principle for external inference must eventually lead back to an iso-regular group. These principles predict the class-inheritance hierarchy of Fig 33, in the following way:

Observe first that, as one descends through the hierarchy, one is reaching successively more symmetrical states. Thus we see that descendence through the class hierarchy is given by the Asymmetry Principle. Furthermore, this downward use of the Asymmetry Principle is given by external inference, and therefore must be realized by the Externalization Principle, which implies that the terminal descendant of each branch must correspond to an iso-regular group. These conclusions are summarized as follows:

THEORY OF CLASS INHERITANCE. *According to the New Foundations:*

> *1. Inheritance is a recovery procedure.*
>
> *2. Descendence through the class hierarchy is given by our fundamental rule of recovery: the Asymmetry Principle.*
>
> *3. Since the recovery is external at each stage, the Externalization Principle applies, and therefore the terminal descendant of each branch corresponds to an iso-regular group.*

According to the New Foundations, this models, rigorizes, and predicts the *discovery* procedure by which software-engineers proceed, as follows: By the principles of good software engineering, one first establishes a base class, and subsequently discovers its descendant classes, in a successive manner, thus ensuring that the code of the base class does not have to be re-written in establishing the code of any of its descendants. What is remarkable is that the New Foundations *predict* the sequence of descendant classes that the software engineer will discover in the development of the software. Thus one will no longer have to wait for the engineer to discover these classes in the usual unguided way.

With this in mind, let us now notice the following: The triangle branch in Fig 33 contains only one TRIANGLE class, and indeed this is not in the form of an iso-regular group. Our theory of inheritance predicts that the software engineer has not actually completed the discovery that will ensue as the software is developed to become more usable by customers. In particular, the New Foundations to Geometry predict that each successive descendant class will remove an asymmetry by external inference. This means that there are two successive classes below the TRIANGLE class: The first is

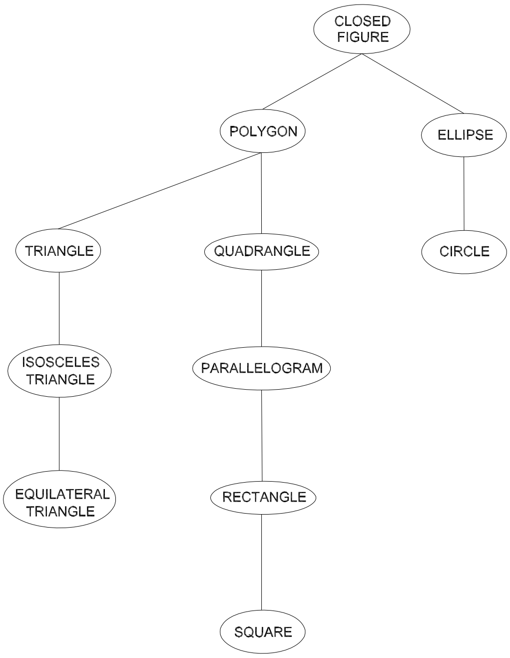

Figure 34: The rigorous class-inheritance hierarchy predicted by the New Foundations to Geometry.

the *isosceles* triangle, and the second and final class is the *equilateral* triangle, which corresponds to the iso-regular group.

Furthermore notice, also by the New Foundations, that the QUADRANGLE class in Fig 33 is missing below it a crucial class, PARALLELOGRAM, which should be inserted between QUADRANGLE and RECTANGLE. Thus, bringing together the conclusions of this and the preceding paragraph, our principles predict that the full class hierarchy which the software engineer will eventually discover is that shown in Fig 34.

27 The *Is-A* Relation

As a result of the above discussion, it is now possible to give a deep algebraic theory of the *is-a* relation that is basic to class-inheritance. As is standardly noted, *is-a* means *sub-class of* rather than *member of*; that is, it really means *is-a-kind-of*. To illustrate the algebraic theory we will develop of this, let us examine the descendant branch starting with the node QUADRANGLE in Fig 34.

One can consider the class text of QUADRANGLE, in the software, as including an invariant stating that there are four sides, and a feature stating that the four side-lengths are real numbers.

Now for the crucial point: Our argument will show that it is fundamentally important to ask what the symmetry group of this structure is. First, we claim that the group is the wreath product:

$$\mathbb{R} \ⓦ \{e\} \quad = \quad [\mathbb{R}_{c_1} \times \mathbb{R}_{c_2} \times \mathbb{R}_{c_3} \times \mathbb{R}_{c_4}] \ⓢ_\tau \{e\}.$$

There are four copies of the fiber $\mathbb{R}$, but the control group $\{e\}$ is trivial: Its action is to leave each side where it is. This corresponds to the fact that, on an arbitrary quadrangle, there is no symmetry group that carries the sides onto each other, because, typically, the sides have different lengths and the vertices have different angles. The *transfer* structure is therefore trivial, and therefore given by $\{e\}$.

Next, move one step down in the class-inheritance hierarchy (Fig 34) to the next node *PARALLELOGRAM*. This class inherits the invariant (four sides) and feature (side-lengths are real numbers) from the class above. However, the symmetry group now increases. It is

$$\mathbb{R} \ⓦ \mathbb{Z}_2$$

where there are again four copies of the fiber, but where the control group as increased to $\mathbb{Z}_2 = \{e\,,\, r_{180}\}$ where r_{180} is 180^0 rotation of the parallelogram (about its center). This is the only isometry that sends the parallelogram onto itself. Notice that all descendants of the quadrangle will have four copies of the fiber, by inheritance.

Now move one step further down the class hierarchy (Fig 34) to the next node RECTANGLE. The symmetry group is still larger, thus:

$$\mathbb{R} \ⓦ [\mathbb{Z}_2 \times \mathbb{Z}_2]$$

where the control group $\mathbb{Z}_2 \times \mathbb{Z}_2$ is the direct product of the reflection group $\mathbb{Z}_2 = \{e\,,\, m_V\}$ where m_V is the vertical reflection, and the reflection group $\mathbb{Z}_2 = \{e\,,\, m_H\}$ where m_H is the horizontal reflection. Notice that the multiple of reflection m_V with reflection m_H is the rotation r_{180} which, by group closure, must also be in $\mathbb{Z}_2 \times \mathbb{Z}_2$. Therefore, the rotation group $\mathbb{Z}_2 = \{e\,,\, r_{180}\}$ of the parallelogram must be a subgroup of the double-reflection group $\mathbb{Z}_2 \times \mathbb{Z}_2$ of the rectangle. In fact, the latter group can be created by what is algebraically called a *group extension* of the former group.

Finally move one step further down the class-inheritance hierarchy (Fig 34) to the bottom node which is the class SQUARE. Here, the symmetry group is larger still:

$$\mathbb{R} \ⓦ [[\mathbb{Z}_2 \times \mathbb{Z}_2] \ⓢ_\tau \mathbb{Z}_2].$$

This extends the group of RECTANGLE by the $\mathbb{Z}_2$ shown on the far right. It is important to notice that the right subsequence $[\mathbb{Z}_2 \times \mathbb{Z}_2] \ⓢ_\tau \mathbb{Z}_2$ is actually the wreath product $\mathbb{Z}_2 \ⓦ \mathbb{Z}_2$ discussed in section 6, and therefore the above group sequence is actually:

$$\mathbb{R} \ⓦ \mathbb{Z}_2 \ⓦ \mathbb{Z}_2.$$

This is an iso-regular group. Therefore, by the New Foundations, it terminates the downward branch.

We have therefore demonstrated this: At each successive class downwards, the symmetry group increases. In fact, the downward hierarchy is given by a sequence of *group extensions*. This information is shown in Table 3.

Quadrilateral	$\mathbb{R} ⓦ \{e\}$
Parallelogram	$\mathbb{R} ⓦ \mathbb{Z}_2$
Rectangle	$\mathbb{R} ⓦ [\mathbb{Z}_2 \times \mathbb{Z}_2]$
Square	$\mathbb{R} ⓦ [[\mathbb{Z}_2 \times \mathbb{Z}_2] ⓢ_\tau \mathbb{Z}_2] \quad = \quad \mathbb{R} ⓦ \mathbb{Z}_2 ⓦ \mathbb{Z}_2$

Table 3: Internal symmetry groups of a class inheritance hierarchy.

Definition 1 *The symmetry group of all the objects of a class will be called the* **internal symmetry group** *of that class.*

The crucial conclusion we have arrived at is this:

THEORY OF THE *IS-A* **RELATION:** *The sequence of* is-a *relationships down any branch of the class inheritance hierarchy corresponds to a sequence of group extensions of the successive internal symmetry groups of the classes, and terminates at an iso-regular group.*

Now, since the internal symmetry group of a class holds for *all* objects of the class, it is an *invariant* of the class. We therefore make a crucial proposal concerning the writing of a class text:

Basic Proposal 2 *The internal symmetry group of a class should be written in the invariant clause of the class text.*

The fundamental consequences of this will now be examined.

28 Object-Creation

Standardly what one means, in object-oriented programming, by the fact that an invariant holds for all objects of a class is that it holds in the following two critical run-time situations: (1) on object-creation, and (2) before and after the remote call of any routine of the class. This section examines the first of these, and sections 29 – 30 examine the second.

In conventional object-oriented programming, the creation procedure of a class produces objects that conform to the invariant clause, in fact, as Meyer [23] p466 says: "a creation procedure's formal role is to establish the class invariant". In contrast, in the form of object-oriented programming we are proposing, the relationship is reversed: The class invariant, as internal symmetry group, *provides* the creation procedure. This is accomplished in the following way:

First, we use a basic principle from Leyton [17]:

SYMMETRY-TO-TRACE CONVERSION PRINCIPLE. *Any symmetry can be re-described as a trace. The transformations defining the symmetry generate the trace.*

Next observe that each of the internal symmetry groups is structured as a hierarchy, in which each level is a direct product of isomorphic one-generator groups, and the relation between levels is given by a wreath product. Now we showed in our books Leyton [16] [17] that any such symmetry group dictates a program, which we call a *canonical plan*, for drawing the figure. It does so in the following way: Each wreath product within the group hierarchy corresponds to a nested do-while loop, where the fiber is a drawing loop, and its control is the drawing loop within which it is nested. The group generator on each level becomes the adder instruction within its corresponding program loop. Note that if the highest level is not transitive in its action on the fiber-group copies immediately below it, then orbits of the fiber-group copies corresponding to the transitive components above are created in parallel. With the correspondences just given between the group products (wreath and direct) and the program structure, the internal symmetry group provides a canonical plan for generating the figure as a trace. Most crucially, we are lead to the following conclusion:

INTERNAL GROUP/CREATION PROCEDURE PRINCIPLE. *The internal symmetry group of a class prescribes, via the canonical-plan realization of the Symmetry-to-Trace Conversion Principle, the creation procedure creating any object in the class as a trace.*

29 Fundamental Structure of a Class

In the type of object-oriented programming we are proposing, the invariant clause contains the internal symmetry group of the class. By the theory presented in this paper, the relation between any command operation and the internal symmetry group is one of transfer. We therefore conclude that this gives a profound structuring of the software text:

FUNDAMENTAL ALGEBRAIC STRUCTURE OF A CLASS: *Each class is given by a wreath product:*

$$G_{Sym} \circledw G(C)$$

where G_{Sym} is the internal symmetry group of the class, and $G(C)$ is the group of command operations.

30 Class Consistency

The new approach to object-oriented programming, that we are proposing, gives a new understanding of class consistency that is far deeper than that which is currently held.

One of the ways we can explain this new understanding is to show that it solves a long-standing problem with respect to the Liskov Substitution Principle (LSP) which states that a routine defined for a base class cannot be violated for any of the latter's derived classes (Liskov [20]). The LSP is desirable because well-designed code is extendable without modifying already-working code; and in the case of class inheritance, this means that the routines of the base class are maintained after adding the latter's descendant classes. Martin [22] [2] showed that the LSP is frequently violated in graphical software, with such examples as follows: If one defines, for the RECTANGLE class, a routine that alters the length of a rectangle object relative to its width, then this would violate the invariance conditions of its child class SQUARE, which include having equal sides. Furthermore, as stated by Meyer [23] p368, the correctness requirement on an exported routine means that executing the body of the routine – started in any state where the class invariant and precondition both hold – must end in a state in which the invariant and postcondition both hold; i.e., the invariant acts as a consistency condition on the *entire* class of objects. Thus the above rectangle routine would violate the invariance condition of its child class SQUARE, and therefore violate the consistency of that class.

We shall now show that this is solved using our approach. The argument takes a number of steps:

First, according to our Internal Group/Creation Procedure Principle (section 28), if one created a rectangle directly from the RECTANGLE class, one would use the internal symmetry group of the class, which Table 3 gives as

$$\mathbb{R} \circledw [\mathbb{Z}_2 \times \mathbb{Z}_2]$$

and the rectangle would be created purely as a trace from the group in the following way: In the above group sequence, the wreath-product sign and the direct-product sign indicate that the four sides are drawn in two reflectional pairs, which implies that adjacent sides are of independent length.

Next suppose that, instead of creating the rectangle directly from the RECTANGLE class, it were generated in the following alternative way that can often occur in a run-time CAD session: First, at some stage in the session, one has created an object from the class SQUARE. Notice therefore that, by our Internal Group/Creation Procedure Principle, the square has been created purely as a trace using the internal group of its class. This group is different from that of the RECTANGLE class; that is, by Table 3, the group is:

$$\mathbb{R} \circledw [[\mathbb{Z}_2 \times \mathbb{Z}_2] \circledS_\tau \mathbb{Z}_2] \;=\; \mathbb{R} \circledw \mathbb{Z}_2 \circledw \mathbb{Z}_2$$

which means that the canonical plan draws a side of only a single length, and *hierarchical transfer* does all the rest.

Then suppose that, later in the CAD session, one has needed to apply the stretch operator to this square object, producing a rectangle. It is at this stage that one apparently

[2] I am grateful to Thomas Patzke of Fraunhofer IESE for introducing me to Martin's paper.

violates the LSP, because the output of the stretch routine has violated the invariant clause of the SQUARE class, i.e., violated what in standard programs is called the equal-sides rule, or what in our system is much more powerfully given as the internal group $\mathbb{R} \circledR \mathbb{Z}_2 \circledR \mathbb{Z}_2$.

However, we will now see how the New Foundations to Geometry solve this problem.

Before we begin, it is worth observing that the displayed figure, the rectangle, can have two *perceptual* interpretations to the viewer: (1) It could be a *rectangle* in the sense that it has been drawn purely as a trace (i.e., internally), or (2) according to our psychological research in Fig 16, it can be viewed as a *stretched square*. These two psychological interpretations therefore correspond to the two interpretations which the viewer can give as to which software class the figure belongs: the RECTANGLE class or the SQUARE class.

It is the second interpretation that corresponds to the apparent violation of the LSP, i.e., the violation of the invariance clause. However, let us now show how our theory solves this apparent violation, both in the software and the corresponding psychological interpretation. It is solved by using our theory of symmetry-breaking in section 12. According to that theory, the reason one can interpret the rectangle as a stretched square is that it is being seen as a *transferred* version of a square onto a rectangle. Thus, under this interpretation, the rectangle is given by the following transfer hierarchy:

$$[\mathbb{R} \circledR \mathbb{Z}_2 \circledR \mathbb{Z}_2] \circledR \textbf{Stretches}$$

where the square is the bracketed part, i.e., the internal symmetry group, and the stretch component is given by the control group to its right. This means that, in conformance with our theory, the square is not lost in its symmetry-breaking, but is transferred onto its symmetry-violating state, the rectangle. Notice that this is embodied in our principle, the Fundamental Algebraic Structure of a Class (section 29), which says that, in a class, the command group is related to the invariant, the square's internal group, via a wreath product. Thus, by transfer – and the algebraic structure which describes it – the class invariant of the square is not lost. Therefore, under our formulation of object-oriented programming, the LSP is not violated.

It is important to understand that this relates to the fact that New Foundations to Geometry are fundamentally different from Klein's Erlanger program, which is the foundation of 20th century mathematics and physics. As the book, Leyton [17], shows in detail: Klein's foundations accord with the conventional view of symmetry-breaking, i.e., the invariants are lost by the groups that break the symmetry actions associated with the invariants. In contrast, by the New Foundations, the symmetry starting states are preserved by the symmetry-breaking actions, due to the recoverability property of our geometry. As a consequence:

NEW THEORY OF INVARIANTS. *According to the New Foundations to Geometry, the invariant is the symmetry ground-state of a generative process. It is an invariant in the sense that all objects of the class possess it by* <u>*recoverability*</u>*. This is represented algebraically by the fact that the symmetry ground-state is a fiber in the symmetry-breaking wreath products that define the objects of the class.*

To illustrate further, let us continue the example of the transition of the square to the rectangle. Following the above approach, one can create the full sequence given by our psychological results in Fig 16, by the following transfer hierarchy:

$$[\mathbb{R} \circledw \mathbb{Z}_2 \circledw \mathbb{Z}_2] \circledw \textbf{Stretches} \circledw \textbf{Shears} \circledw \textbf{Rotations} .$$

This is interpreted as the upward transfer of the square onto the rectangle – which is thereby structured as a stretched square – the latter being then transferred onto the parallelogram – which is thereby structured as a stretched sheared square – the latter being then transferred onto the rotated parallelogram – which is thereby structured as a rotated stretched sheared square.

With this analysis, we are lead to a crucial conclusion: While the downward class inheritance hierarchy is such that an ancestor class does not "know" anything about the structure of its descendants, one can, at run-time, reconstruct the hierarchy such that an object on the ancestor level can be re-interpreted as the successive upward transfer of descendants. This process works exactly because it is an example of our theory of *recoverability*, the fundamental rule of which is the Asymmetry Principle, which states that the only recoverable operations are symmetry-breaking ones, and the further rule that all external uses of the Asymmetry Principle conform to our Externalization Principle which states that external inference leads ultimately to a past state whose internal structure is an iso-regular group. What the above run-time upward-reconstruction of the hierarchy does is to externalize an object at any ancestor level as a sequence of symmetry-breaking transfers upward through the descendants, the lowest of which is the recovered iso-regular group.

As a result, we propose a new object-oriented operator based on this principle.

EXTERNALIZATION OPERATOR. *Given an object on any level, convert it into the sequence of upward symmetry-breaking transfers of the recovered objects from its descendant classes. The use of the externalization operator will be said to be* full *if the selected starting descendant level is that given by the* iso-regular group *recovered from the object.*

The above operator tells us how to move from the abstraction hierarchy to the corresponding concrete generative hierarchy. The reverse is also possible. Thus, in prototype-based programming [35] [7], which is a concrete generative process without classes and abstraction, our theory gives a principled means of producing classes from concrete generative sequences.

In fact, as explained in my book *Symmetry, Causality, Mind* (MIT Press), a crucial aspect of the New Foundations to Geometry is that it uses a prototype approach. Furthermore, the New Foundations is the only system that gives a mathematical theory of prototypes and their hierarchical relation to the other objects. Also, as shown in my book, *A Generative Theory of Shape* (Springer), this mathematical theory is fundamentally the opposite of the Standard Foundations.

31 Maximization of Reusability

According to the New Foundations to Geometry:

The reusability of an object is maximized if the object itself is defined as having been produced by maximizing reuse of the operations that were used to produce it.

This is because reuse *within* the object achieves most of the reuse that is needed when the entire object has to be reused.

Therefore the object must be represented generatively; and the generative operations used to represent it must be maximally reused in that representation.

> **For example:**
>
> **Given a science data set, the New Foundations to Geometry give the data set a generative representation in which *all* levels of the data set are generated by reuse.**
>
> **Also, the New Foundations to Geometry give software a generative structure in which *all* levels of the software are generated by reuse.**
>
> **Also, the New Foundations to Geometry give any design a generative structure in which *all* levels of the design are generated by reuse.**

In the New Foundations to Geometry, maximization of reuse of the generative operations is called Maximization of Transfer.

Now, to ensure the maximization of reuse of the generative operations, the operations must be maximally *recoverable.*

Therefore, the maximization of reuse is dependent on the maximization of recoverability.

Therefore, according to the New Foundations to Geometry, in order to ensure maximization of reusability of an object, it must be given a representation that accords with the two basic principles: Maximization of Transfer and Maximization of Recoverability. Therefore, we conclude the following:

FUNDAMENTAL LAW OF PERSISTENT REUSE

Persistent reuse of an entity, over the data lifecycle and product lifecycle, is achieved by defining the entity *generatively* **such that it accords with the two basic principles of the New Foundations to Geometry: Maximization of Transfer and Maximization of Recoverability of the generative operations.**

References

[1] Allen, G. (2000). CAD/CAM Data Interoperability Strategies. In W. Dankwort & J. Hoschek (Editors) *Digital Products*. Stuttgard: B.G. Teubner, 7-25.

[2] Blum, H., (1973). Biological shape and visual science. *Journal of Theoretical Biology*, **38**, 205-287.

[3] Capoyleas, V., Chen, X. & Hoffmann, C.M (1996). Generic naming in generative, constraint-based design. *Computer-Aided Design*, **28(1)**, 17-26.

[4] Costa, L. da F. (2000). Robust skeletonization through exact Euclidean distance transform and its application to neuromorphometry. *Real-Time Imaging*, **6**, 415-431.

[5] De Sa, R., Radice, G., & Kerckhove, M. (1999). Timing and 3-D analysis of musculoskeletal development. NSF Final Report. Univesity of Richmond.

[6] Deguchi, K. & Furukawa, R. (2001). Curvature-based diagram description of deforming contour and its application to X-ray Cardiac Image Diagnosis. *Interdisciplinary Information Sciences*, **7 (2)**, 197-208.

[7] Dony, C., Malenfant, J. & Cointe, P. (1999). Classifying prototype-based languages. In J. Noble, A. Taivalsaari, I. Moore (Editors) *Prototype-Based Programming Languages*. Berlin: Springer-Verlag

[8] Kirupaharan, N. & Dayawansa, W.P. (2001). Theory of reference frames and biological control. *Mathematical and Computer Modeling*, **33(1-3)**, Jan-Feb, 193-198.

[9] Larsen, T.W. (1993). Proces grammatik og proces historie for 2D objekter. DAIMI IR-115, Aarhus Univ.Report

[10] Leyton, M. (1986a). Principles of information structure common to six levels of the human cognitive system. *Information Sciences,* **38**, 1-120. Entire journal issue.

[11] Leyton, M. (1986b). A theory of information structure I: General principles. *Journal of Mathematical Psychology,* **30**, 103-160.

[12] Leyton, M. (1986c). A theory of information structure II: A theory of perceptual organization *Journal of Mathematical Psychology,* **30**, 257-305.

[13] Leyton, M. (1987b) Symmetry-curvature duality. *Computer Vision, Graphics, and Image Processing,* **38**, 327-341.

[14] Leyton, M. (1988) A process-grammar for shape. *Artificial Intelligence,* **34**, 213-247.

[15] Leyton, M. (1989) Inferring causal-history from shape. *Cognitive Science,* **13**, 357-387.

[16] Leyton, M. (1992). *Symmetry, Causality, Mind.* Cambridge, Mass: MIT Press.

[17] Leyton, M. (2001). *A Generative Theory of Shape.* Berlin: Springer-Verlag.

[18] Leyton, M. (2006) Interoperability and objects. *International Journal of Product Lifecycle Management,* **1 (2)**, 98-128.

[19] Lin W-C., Liang C-C. & Chen C-T. (1989). A computational model for process-grammar. *Artificial Intelligence*, **38** , 207-224.

[20] Liskov, B. (1988). Data abstraction and hierarchy. *SIGPLAN Notices,* **23**, 5.

[21] Lopéz, A.M. (1997). Ridge/valley-like structures: creases, separatrices, and drainage patterns. http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/LOPEZ/evonline.html

[22] Martin, R.C. (1996). The Liskov Substitution Principle. Report available at http://www.objectmentor.com/resources/articles/lsp.pdf

[23] Meyer, B. (1997). *Object-Oriented Software Construction*. New Jersey: Prentice Hall.

[24] Mayoh, B. (1995). On patterns and graphs. Technical Report DAIMI PB 484, Aarhus University, Comp.Sci.Dept.

[25] Milios, E.E. (1989). Shape matching using curvature processes. *Computer Vision, Graphics, and Image Processing*, **47**, 203-226.

[26] Ogniewicz, R.L. (1995). Automatic medial axis pruning by mapping characteristics of boundaries evolving under Euclidean geometric heat flow onto Voronoi skeletons. Harvard Robotics Laboratory Technical Report No. 95-4.

[27] Parvin, B., Peng, C., Johnston, W. & Maestre. M. (1994). Tracking of Tubular Objects for Scientific Applications. *IEEE Conf. on Computer Vision and Pattern Recognition*, 295-30.

[28] Pernot, J-P., Guillet, S., Leon, J-C., Falcidieno, B., & Giannini, F. (2003). Interactive Operators for free form features manipulation. In SIAM conference on CADG, Seattle, 2003.

[29] Pizer, S.M., Fritsch, D.S., Yushkevich, P.A., Johnson, V.E., & Chaney, E.L. (1996). Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging*, **18(10)**, 851-865.

[30] Pratt, M. (2004). Extension of ISO 10303, the STEP standard, for the exchange of procedural shape models. In *Proceedings: Shape Modeling International (IEEE).* Genova, Italy.

[31] Product Engineering Program, Manufacturing Systems Integration Division, NIST. http://www.mel.nist.gove/msid/pe.htm

[32] Richards, W., Koenderink, J.J., & Hoffman, D.D. (1987). Inferring three-dimensional shapes from two-dimensional silhouettes. *Journal of the Optical Society of America A*, **4**, 1168-1175.

[33] Shemlon, S. (1994). *The Elastic String Model of Non-Rigid Evolving Contours and its Applications in Computer Vision*. PhD Thesis, Rutgers University.

[34] Torres, R da S, & Falcão, A. X. (2006). Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, , 1–11.

[35] Ungar, D. & Smith, R.B. (1987). Self: The Power of Simplicity. In *Object-Oriented Programming Systems, Languages, and Applications.* ACM Press, 227-242.