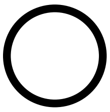
Laws of Form as Avatar Extension

by Sven Nilsen, 2024

In this paper I explain Laws of Form from the perspective Avatar Extensions and use it to prove that logic emerges from Form using a 1-avatar and the two interpretations of space as collection and enumerable structure. I discuss the difference between constructive and classical logic.

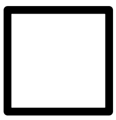
Laws of Form[1] is a book written by George Spencer-Brown in 1969. The idea is to operate on a level of language prior to symbolism[2] where meaning is grounded in the self-reference of making distinctions, stating multiple propositions[3] at once in the relationship between the parts.

To explain this philosophy, consider drawing a circle:



When I show this circle to somebody, I do not operate on the level of symbolism where the circle needs to mean something. To do so would be to add some proposition[3] about the circle that has a separated meaning from other propositions that appear at once in the relationship between the circle, its inner region and outside of the circle. Only in the sense of drawing a circle I can express all these propositions at once in the relationship between the parts.

Now, consider drawing a square:



In my use of this language, I will treat the square in the same way as the circle under all circumstances, being able to deform a circle into a square dynamically. Only the topology[4] of the drawing is preserved. This axiom is introduced without ever leaving the level of language prior to symbolism. Preservation of the topology here is analytic[5], but I will later show how one can reference a such activity from within symbolism that preserves the topology in a synthetic[5] sense.

When referencing this language, I call it Form[1], which in general can be thought of as the activity of drawing, but also takes on a dual meaning of the specific examples I mentioned above. I use a symbol to refer to some specific drawing. Form can also through the dual meaning represent drawing in general as an activity:

Now I am operating on the level of language within symbolism. Yet, the symbol itself is formed to be self-referential to the activity of drawing. I can use the activity of drawing to reference, or describe, the activity of drawing. The purpose is to break down the language barrier between symbolism and the level of language prior to symbolism. This way, this form of communication becomes a way of giving instructions to the reader, where these instructions are both topological in the sense of analysis and synthesis. The idea is to say more with less building blocks.

Now, I will use 1-avatars[6] and language bias analysis to explain what is happening.

Let `a` be some object that has a role[7] `p`, written (using notation Avatar Logic[7]):

a : p

An object can only have one role, which in many situations lead to some over-constrained language. With other words, the limitation of only having a single role puts so much pressure on the use of language that the language itself is often not powerful enough to express a problem sufficiently.

A 1-avatar `b` is a way of extending `a` such that the two, together, can take on a new role `q`:

b’(a) : q

1-avatars relaxe the language from being over-constrained[8].

The opposite approach of using 1-avatars is by using a language that is under-constrained or unconstrained (e.g. First-Order Logic[9]). One adds axioms to constrain the language, e.g. when I previously went from the activity of drawing in general to the topological interpretation of drawing.

Technical comment:
In First-Order Logic, it is common to start with predicates, which are unconstrained.
To describe a problem sufficiently, one adds axioms about the predicates in question.
This can lead to not being able to talk about uniqueness in a pragmatic way.
Avatar Logic is an alternative to First Order Logic where uniqueness is more natural.

The Form in sense of some specific drawing, is a 1-avatar:

a : p ↴’(a) : q

Before drawing anything, one starts with an empty canvas:

(empty canvas)

The infinite empty canvas, without limits, has the property that it looks the same no matter how you orient it, translate it or skew it, or deform it, as long the transformation preserves the cover of space.

When drawing, one is creating a distinction between two parts of space, the two sides of the line:

↴

This introduces a limit to the space, that constrains which transformations one can do, while keeping the drawing look the same way. For many applications, this is an over-constrained language, so I relax the language by focusing in on the topological properties. Since Form references the activity of drawing under topology, I use it as a 1-avatar to give space the possibility of taking on some new role. This new role holds under the topological transformations.

Now, to describe which part of space I allow taking on a new role, I need to distinguish between the parts, the two sides. Symbolically, I call the left side for the Inside and the right side for the Outside:

Inside ↴ Outside Inside : p ↴’(Inside) : q

The Outside preserves the properties of space in general in every way, except the difference that has been made by introducing some Inside distinct from it. I am subtracting from the Outside by Inside.

To understand Form, one can approach it from the perspective of Inside, which is synthetic, or from the Outside, which is analytic. Approaching Form does not imply that I am crossing over to the other side. It merely means to approach the boundary, to get closer, to the topological limit of space.

In the level of approaching some limit of language one can understand, how it makes sense to stay within some specific use of rules: The rules constrain the language from being unified with language as a general activity, while serving the purpose of approaching truth, without being truth.

Since Form functions like a 1-avatar that allows space to take on a new role in the Inside, while subtracting this meaning from the Outside, it means that Form belongs on the side of Inside. While biasing Form toward the Inside, this simultaneously unbiases it from the Outside, such that the Form that is well known from the Inside becomes an intrinsically unknown in the Outside.

The property of Form being well known from the Inside, has significant implications in the philosophy of mathematical language design. When assuming that this is the only property that distinguishes Inside from Outside, one can by induction infer that Inside languages, which are languages biased toward Inside, are unified in the sense of well-knownness. This give rise to the distinction between Inside vs Outside theories[10] of mathematics (from Avatar Schema Theory[10]):

- An unknown in an Inside theory is some sentence in a grammar
- All Outside theories contain at least one symbol which is intrinsically unknown

With other words, the reference to Form in Outside theories is a reference to the unknown.

Technical comment:
When something is described with Inside theory in full detail,
one no longer needs the Outside.
For example, by describing a car in full detail,
every property of the car can be reasoned about by using a computer to simulate the car.

Yet, there is a sense of the car which is not captured by describing it in full detail,
which is the very purpose of having a car in the first place: Personal transportation.
This property is only possible in Outside theories.

Personal transportation has to be unknown in some sense,
since the role it plays in language can not be fully described.
When I use a car, I am applying Outside theory,
where the car functions as an Avatar Extension to my own personal autonomy.

Now, I will introduce The Law of Calling[1]:

↴↴ = ↴

The Law of Calling operates on the level of language where space is interpreted as a collection. Specifically, The Law of Calling constrains collections of drawings to have a Set-like property. With other words, if I make two drawings that are topologically equivalent to each other, in every circumstance I treat these two drawings as the same as a single drawing of either object.

Propositionally, The Law of Calling gets translated in logic in two ways that are dual to each other:

a & a <=> a a | a <=> a

However, when Inside is `a` and one includes the Outside as `b`, there are 3 ways of interpretation:

a & a & b   <=>   a & b
a | a | b   <=>   a | b
a => (a => b)   <=>   a => b

Laws of Form[1] favors the third interpretation using `=>`.

In classical logic, `=>` can be defined using OR and NOT:

a => b   <=>   !a | b

In constructive logic[11], `a => b` is different from `!a | b`, which only becomes unified by adding excluded middle `a | !a` for all `a`. One has to be careful to not confuse the two constructively. Therefore, when using constructive logic I will be explicit of whether Form uses `a => b` or `!a | b`.

The reason Laws of Form favors `=>`, is because Form is biased toward the Inside. This allows Form to take on the role of Negation in the sense of `!a | b`. This Negation is an integrated part of the binary operator, but it is operating on the left argument.

Negation is not the same as Form in general, but in the specific case where Form allows space to take on a new role as the Inside, these two become one and the same. Negation is also a 1-avatar.

It is not always the case that one can introduce new 1-avatars. Consider the following:

!!a   <=>   a

This theorem holds in classical logic, but not in constructive logic[11].
In constructive logic one can prove `a => !!a`, but not `!!a => a`.

Now, if `a` can be replaced by something, called `b`, that is negated, then I can write the following:

!!!b   <=>   !b

This holds in constructive logic[11]. So, if `a` is a Negation of some `b`, then it behaves classically.

One design principle in Laws of Form is that Form is the only fundamental symbol, in order to play with recursion and self-referential paradoxes. This principle biases Laws of Form toward classical logic, which might have been the intention of the author. However, by understanding Form as an Avatar Extension, one can treat Laws of Form as a general framework where the over-constrained bias toward classical logic can be relaxed to constructive logic.

Constructively, The Law of Calling has two interpretations in logic:

a => (a => b)   <=>   a => b
!a | !a | b   <=>   !a | b

This means, I do not have to worry about which interpretation I am using when applying this axiom:

a ↴ (a ↴ b)   <=>   a ↴ b

I can use the symbol for Form, `↴`, to refer to either interpretation.

Negation can be described as following:

a ⮧ false

In constructive logic[11], it is common to define `!a` as `a => false`.
This means than when using `!a | b`, one can write it as `(a => false) | b`.
Since `a | false <=> a` for any `a`, one gets `a => false` from `(a => false) | false`.
This shows that `a ⮧ false` is equivalent to Negation regardless of interpretation.

Some logicians define `false` as `a & !a`, which in Laws of Form would be like `false(a)`.
In that case one needs `false(a) <=> false(b)` for all `a, b`. I will not deal with this issue here.
The space of `false` might be thought of as empty, that is without any interior.
Since the Inside space is given a new role by Form, this gives a perspective of the Inside as being Negated. Without any interior and the bias of Form toward the Inside, `false` in this context refers to the Inside from the Outside. So, `false` in this sense is an intrinsically unknown symbol.

Negation in this way is suppressing the Inside, not by denying it explicit reality, but by pushing it outside of the domain of knowledge, of well-knownness, from the perspective of language.

This intuition supports the idea that `false` can imply anything, the absurdity property of `=>`:

false ⮧ a	can be introduced everywhere for every `a`
false => a	absurd
!false | a	`!false <=> true` and `true | a <=> true`

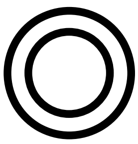
From this one can derive that `false ⮧ a` implies some `true` that can be introduced everywhere.
The duality between `false` and `true` is a consequence of the symmetry between absurdity and the ability to introduce `!false | a` everywhere.

Notice that I play on the constructive ambiguity of Form above to derive properties of logic.

Now, I will introduce The Law of Crossing:

⮧^⮧ =

Here, I use `^` to signify the drawing of a circle inside a circle without including anything else:

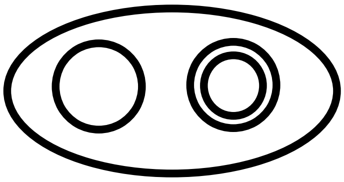
 =

Technical comment:
Constructively, the `^` symbol is a hidden exponential that restricts `=>`to some exclusive context.

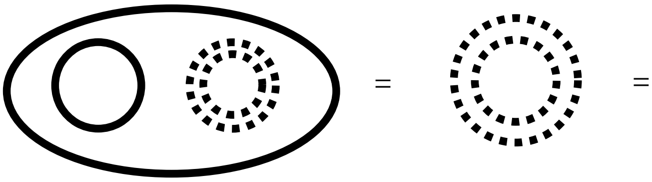
It means `a^b` implies `b => a`, but without any extra context than `b`.
Another way to write `a^b` is `(b => a)^true` or `b → a`.

In modal logic, one might write `□(b => a)`,
but this should not take on the meaning of Provability Logic since Löb's axiom is absurd.
Instead, it is preferrable to use `^` as a binary operator and add the HOOO EP axioms.

For example, when there are two distinct objects inside a circle, this is seemingly irreducible:

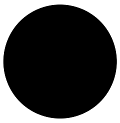


You might see the problem: By applying The Law of Crossing twice, I can erase the drawing:



This means in the language of only drawing circles, there is no way to express anything else than simply nothing, or a circle. This holds for any shape: You can only express the shape or nothing.

A way to work around this limitation is by nesting the circles inside each other infinitely number of times, so that no matter how many Crossings are being made one can not reduce it. Every Crossing leaves an infinite number of circles nested inside each other and one gets back to where one started. This infinite repetition can be represented as a solid disc:



Now, due to the topological interpretation, this solid disc is equal to a point.

By approaching Form through either perspective of the Inside or Outside, one interprets a point as a part of space where the Inside only consists of Form. A point is just a point, set apart from everything else. An unlabeled point is pure Form. When labeled, it can be given a role[7], which is defined in relation to everything else that the point is not, the space:

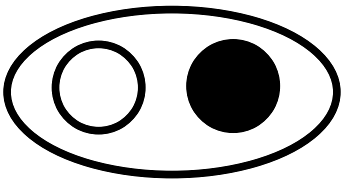
p : T `p` is a point, `T` is a space (also thought of as a role or type)

The unlabeled point is simply Form itself, as a thing-in-itself, which is contractible where it can be:

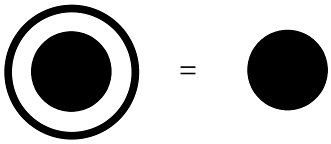
↴

Previously, I talked about Form as a 1-avatar, which is a kind of unary operator, or a relation between Inside and Outside, which is a kind of binary operator. Now, Form also gets a third interpretation which is something on its own. The activity of drawing is a kind of time-less activity that might be seen as happening in an instant, a single moment, or equivalently, lasting for eternity.

The simplest irreducible object (besides a point itself) is a circle with one circle and a point:



If I have a circle with just a point inside, then naturally this is just a point:



The Law of Crossing might be written this way:

a ↴^↴ b = a b `a b` is either `a & b`, or more commonly used as `a | b`

A space containing a point `a` and some `b`, written `a b`, is indistinguishable (which means that in every circumstance I treat it the same way), from a space where I draw circles around the point `a`.

However, if `a` is a circle, then when I draw a circle around `a` it erases `a` from the space.
This leaves behind only `b`, which is not the same as `a b` when `a` is a circle.
From this interpretation one can also see that drawing a circle is a form of Negation.

In the theory of Avatar Extensions[6] there is another construction that is neither a circle nor a point:

The 0-avatar.

A 0-avatar is empty inside. It has neither a space inside, like a circle, nor an infinite recursion of circles, like a point. It is impossible to draw a 0-avatar on paper. Hence, a 0-avatar might be thought of as existing "outside" the very activity of drawing itself. It is neither in the Inside nor the Outside.

While Avatar Extensions is an Outside theory and draws 0-avatars as nodes in Avatar Graphs[12], it treats the 0-avatar as the symbol that is intrinsically unknown to its theory.

Now, since Form is the 1-avatar and the space is empty inside, but given a new role:

false : _ ↴'(false) : _

One can think of the 0-avatar as an arbitrary complex Form where it relates to an Empty Inside:

false ↴ 0-avatar by Empty Inside, Form can control the absurdity arbitrarily

This makes the 0-avatar having the power of absurdity on the Outside, but controlled by Form.
It is modelling lying in logic. This leads to the concept of Abstract Corruption[13][14] in Avatar Extensions[6].

false ↴ a can be introduced everywhere

For the absurdity property, see earlier in the paper, where I show this holds constructively in both interpretations of `a ↴ b` as either `a => b` or `!a | b`.

Using the Tetralemma/Catuṣkoṭi[15], one can give the following table (that is interpreted vaguely):

Catuṣkoṭi	**Laws of Form**	**Topology**	**Truth Value**	**Computation**	**Avatar Extensions**	**Algebra**
A	↴^∞↴	point	true	Infinite recursion	Contractible natural 1-avatar(s)	1*x = x*1 = x
!A	↴ false	circle	false	Empty Outside	1-avatar	-x, x* **i**
A & !A	(↴ false | ↴^∞↴)↴	structure	both	Irreducible	2-avatar	x + y, x * y
!(A | !A)	false ↴	hole	neither	Empty Inside	0-avatar/core candidate	0*x = x*0 = 0

References:

[1] "Laws of Form"
Wikipedia, George Spencer-Brown, 1969
https://en.wikipedia.org/wiki/Laws_of_Form

[2] "Symbol"
Wikipedia
https://en.wikipedia.org/wiki/Symbol

[3] "Proposition"
Wikipedia
https://en.wikipedia.org/wiki/Proposition

[4] "Topology"
Wikipedia
https://en.wikipedia.org/wiki/Topology

[5] "Analytic-synthetic distinction"
Wikipedia
https://en.wikipedia.org/wiki/Analytic%E2%80%93synthetic_distinction

[6] "Avatar Extensions"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html

[7] "Avatar Logic"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html#avatar-logic

[8] "Introduction: The Problem of Over-Constrained Theories"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html#introduction-the-problem-of-over-constrained-theories

[9] "First Order Logic"
Wikipedia
https://en.wikipedia.org/wiki/First-order_logic

[10] "Avatar Schema Theory"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html#avatar-schema-theory

[11] "Intuitionistic Propositional Logic"
Stanford Encyclopedia of Philosophy
https://plato.stanford.edu/entries/logic-intuitionistic/

[12] "Avatar Graphs"
AdvancedResearch – Summary page on Avatar Extensions
https://advancedresearch.github.io/avatar-extensions/summary.html#avatar-graphs

[13] "Abstract Corruption"
Sven Nilsen, 2023
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip2/abstract-corruption.pdf

[14] "Abstract Corruption"
AdvancedResearch – Summary page on Avatar Extensions
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip2/abstract-corruption.pdf

[15] "Catușkoți"
Wikipedia
https://en.wikipedia.org/wiki/Catu%E1%B9%A3ko%E1%B9%ADi