Combinatory Logic and the Laws of Form

The calculus of indication as a special combinatory logical system

Rudolf Kaehr Dr.phil@

Abstract

A comparatistics of calculi, formal systems or algorithmic languages should compare the different approaches involved and highlight common and different features. This is well done with the studies of combinatory logic, lambda calculus, category theory and logic. There is yet no similar comparatistics study concerning more deviant systems like the calculus of indication of the Laws of Form (G. Spencer-Brown). This sketch takes a first step to analyze the Laws of Form (FoL) in the context of Combinatory Logic (CL). A first result of a translation of the LoF into the framework of the CL shows the definability of the LoF initial *Order* by the initials *Number* and *Transposition* only. (Work in progress, v. 0.3)

1. Comparatistics

1.1. Combinatory Logic and the Laws of Form

1.1.1. Questions of comparing the LoF with other calculi

"Jeff James (1993) has developed a system, inspired by Spencer-Brown's work, which does arithmetic using a topological notation with only two kinds of boundary. These may be represented textually as brackets () and [], with the interpretation of the empty expression as zero, () as one, ()() as two etc., addition is simple juxtaposition ab and multiplication is ([a][b]) and so on. If () is understood as an exponential function and [] as a logarithmic function, it all suddenly makes sense, however the arithmetic is actually performed using only four simple axioms: ([a]) = a, [(a)] = a, []a = [], ([a][b])x = ([ax][bx]).

I mention this because I can't help feeling that it is somehow related to the lambda calculus and combinators but I don't see how."

David C Keenan
http://dkeenan.com/Lambda/

Due to the lively feedback I got for my last papers about graphematics and the *Laws of Form* I feel motivated to continue some further analysis of the intriguing features of both approaches to formalization.

Instead of involving the CI into a *complementary* context, like the newly introduced calculus of differentiation, another strategy to get rid of the *indicational nightmare* is recommended: show its similarity or even isomorphy, and differences too, to other much more familiar calculi.

It is sketched without much comments that the CI has some isomorphic neighbors like *Combinatory Logic*, CL, *Category Theory*, CAT, *Boundary Arithmetic*, BA, all together more abstractly conceived than Propositional Calculi, PC, and therefore, probably less well known.

Each approach represents a different thematization of the conception of calculi and computability with its own special properties, despite of some kinds of abstract resemblance or even isomorphism.

Epistemologically and from a meta-theoretic point of view there are significant similarities between the CI and the CL to detect. In comparison with propositional logic, those CL similarities are much closer to the intentions of the CI. Both are emphasizing a kind of an independence from the strict operator/operand-dichotomy. Combinators in CL are playing both role, operator and operand. In a similar sense, the *Laws of Form* are insisting on the double-function of the cross, being operator and operand at once.

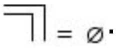
The main difference between the CI and the CL seems to be the CI-abstraction from the commutativity, i.e. the special *topology-invariance,* of its terms as an underlying presumption of the CI, while the CL is strictly build on the identity of its terms.

Further questions arises: If there is a complementary calculus to the calculus of indication, what is the complementary calculus to the Combinatory Calculus?

Again, this is just a preliminary sketch, and further elaborations are left to the enthusiastic reader.

It is well known and often played as a sophisticated sport that it is worth to try to reduce the number of axioms of the original axiom systems and to proof its completeness despite the loss of an axiom of the axiom system as it was presented originally by its inventor.

This holds prominently for the CL, where it is now common praxis to not to use the identity axiom, **I** x = x, because of its derivabiltity from the other axioms, i.e. combinators, like **K** and **S**.

A similar reduction might happen with CI's Primary Arithmetic axiom of *Order*, .

Identity seems to be trivial for the CL because the whole concept of formalization of the CL holds in the framwork of identity. Non-identical systems are covered by the scriptures of the graphematic system, and a formalization then demands identity rules on different levels.

Some questions

Is it possible to define in CI the Initial arithmetic *Order*, I2, with the Initial arithmetic *Number*, I1, and the algebraic *Transposition*, J2, only?
As a consequence: Is it possible to define the CI axioms with *number*, I1 and *distribution* (transposition) J2 only?
Position, J1, appears then as a specified Order, I2.

Strategy

Initials in the CI are considered by GSB as *imperatives* and not as axiomatic relational statements. Hence, the initials can be abstractly understood as *operators*. Therefore, the name of the imperative indicates an abstraction from the arrangement of the imperative and deserves an operator, i.e. a combinator in the sense of Combinatory Logic.

Combinatory Logic, CL, studies the process of substitutions. For the CI, substitutions are not reflected but considered as not specially problematic. Therefore, substitutions are used to connect the primary algebra with the primary arithmetic in the CI.

With the proposed translation or modeling of the *Laws of Form* to a system of

Combinatory Logic, all the techniques and tools, like programming environments and meta-theoretical analysis, should then be available for further studies of the Brownian universe of distinctive arrangements.

Recalling some conventions about operators and operands in generall

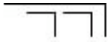
"An expression may occur in three positions as a component of a larger expression:
1. in the operator *position,*
2. in the operand *position,*
3. as the body *of another lambda expression.*
The lambda expression is the second basic method of assembling a new expression.
In their most austere form the expression under consideration may be characterized as
follows.

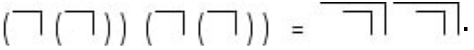
An expression is
   either simple and is an identifier
   or a lambda expression
      and has a bound variable which is an identifier
      and a body which is an expression,
   or it is composite
      and has an operator and an operand, both of which are expressions.

A rule is needed for recognizing when the body of a lambda expression ends. The rule
is that the body extends as far as it can until it is terminated by a closing bracket, com-
ma, or the end of the whole expression. It follows that parenthesis are only needed to
enclose the body if it is a list although they may be used if this improves readability."
W.H. Burge, Recursive Programming Techniques, 1975, p. 9
http://www.thinkartlab.com/pkl/lola/poly-Lambda_Calculus.pdf

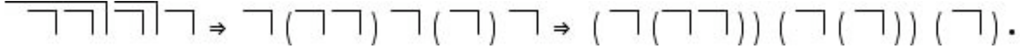
1.1.2. Operational notation of CI-expressions

Also there is much emphasis on the neutrality of the cross as being both an operator and an operand, an operational notation makes it clear that a permutation of an operator with an operand is producing a strictly different expression, which even might turn out to be a non-expression in the CI. Hence, the distinction of *operator* and *operand* is nevertheless crucial for the CI too.

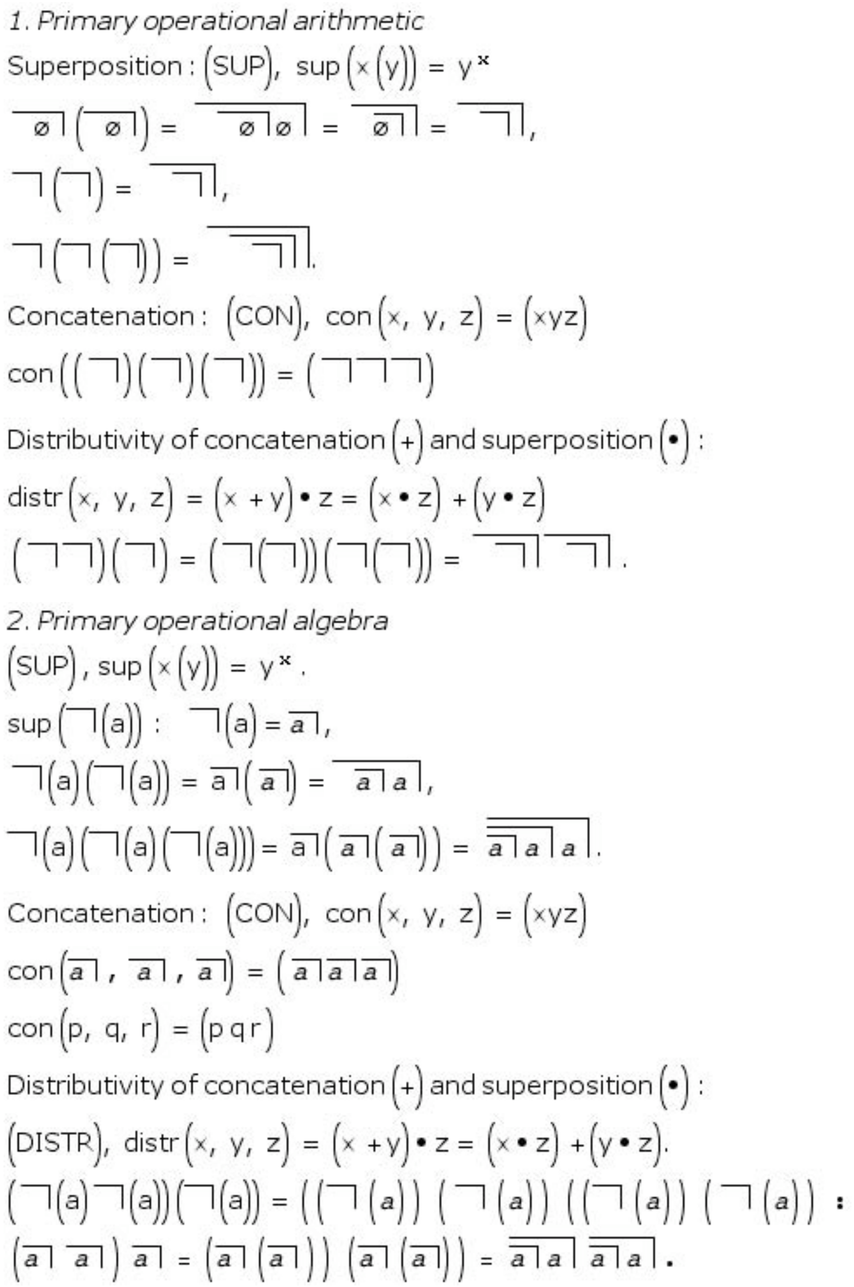
Example
Take, ┐(┐┐) !=$_{\mathrm{CI}}$ (┐┐)┐.
"┐(┐┐)" ∈ CI as , while "(┐┐)┐" has no proper representation in the CI.
By definition, the expression, (┐┐) (┐), or (┐)(┐┐) , gets a representation in an operational approach of the CI as:



Transcription:



Definitions



1. Primary operational arithmetic

Superposition : (SUP), $\mathrm{sup}(x(y)) = y^{x}$

$\overline{\emptyset}\rceil(\overline{\emptyset}\rceil) = \overline{\overline{\emptyset}\rceil\emptyset}\rceil = \overline{\overline{\emptyset}\rceil}\rceil = \overline{\overline{\ }\rceil}\rceil$,

$\overline{\ }\rceil(\overline{\ }\rceil) = \overline{\overline{\ }\rceil}\rceil$,

$\overline{\ }\rceil(\overline{\ }\rceil(\overline{\ }\rceil)) = \overline{\overline{\overline{\ }\rceil}\rceil}\rceil$.

Concatenation : (CON), $\mathrm{con}(x, y, z) = (xyz)$

$\mathrm{con}((\overline{\ }\rceil)(\overline{\ }\rceil)(\overline{\ }\rceil)) = (\overline{\ }\rceil\overline{\ }\rceil\overline{\ }\rceil)$

Distributivity of concatenation (+) and superposition (•) :

$\mathrm{distr}(x, y, z) = (x + y) \bullet z = (x \bullet z) + (y \bullet z)$

$(\overline{\ }\rceil\overline{\ }\rceil)(\overline{\ }\rceil) = (\overline{\ }\rceil(\overline{\ }\rceil))(\overline{\ }\rceil(\overline{\ }\rceil)) = \overline{\overline{\ }\rceil}\rceil\,\overline{\overline{\ }\rceil}\rceil$.

2. Primary operational algebra

(SUP), $\mathrm{sup}(x(y)) = y^{x}$.

$\mathrm{sup}(\overline{\ }\rceil(a))$: $\overline{\ }\rceil(a) = \overline{a}\rceil$,

$\overline{\ }\rceil(a)(\overline{\ }\rceil(a)) = \overline{a}\rceil(\overline{a}\rceil) = \overline{\overline{a}\rceil a}\rceil$,

$\overline{\ }\rceil(a)(\overline{\ }\rceil(a)(\overline{\ }\rceil(a))) = \overline{a}\rceil(\overline{a}\rceil(\overline{a}\rceil)) = \overline{\overline{\overline{a}\rceil a}\rceil a}\rceil$.

Concatenation : (CON), $\mathrm{con}(x, y, z) = (xyz)$

$\mathrm{con}(\overline{a}\rceil, \overline{a}\rceil, \overline{a}\rceil) = (\overline{a}\rceil\overline{a}\rceil\overline{a}\rceil)$

$\mathrm{con}(p, q, r) = (p\,q\,r)$

Distributivity of concatenation (+) and superposition (•) :

(DISTR), $\mathrm{distr}(x, y, z) = (x + y) \bullet z = (x \bullet z) + (y \bullet z)$.

$(\overline{\ }\rceil(a)\overline{\ }\rceil(a))(\overline{\ }\rceil(a)) = ((\overline{\ }\rceil(a))\ (\overline{\ }\rceil(a))\ ((\overline{\ }\rceil(a))\ (\overline{\ }\rceil(a))$ **:**

$(\overline{a}\rceil\ \overline{a}\rceil)\ \overline{a}\rceil = (\overline{a}\rceil(\overline{a}\rceil))\ (\overline{a}\rceil(\overline{a}\rceil)) = \overline{\overline{a}\rceil a}\rceil\ \overline{\overline{a}\rceil a}\rceil$ **.**

The algebraic structure of the operational LoF scripture

The syntactic rules of the LoF scriptures are not the rules of the calculus "out of the syntax". The LoF-scripture is constructed by the unary superposition operation (•) and the binary concatenation operation (+) over the base $B = \{\overline{\ }\rceil, \emptyset\}$.

Definition

Cooperation (•): $\overline{a}\rceil b = a \bullet b$.

Superposition (•): $\overline{a}\rceil\overline{\ }\rceil = \overline{\ }\rceil \bullet (a) = \overline{\ }\rceil (a) = \overline{a}\rceil$.

Coalition (+): a + b = (a b) : Concatenation

LoF – script

$\text{LoF-syntax} = (\text{lof}, \{\urcorner, \varnothing\}, (+), (\bullet), =)$

with

concatenation : $(+)$ and *superposition* : $(\bullet)$

commutativity : $x + y = y + x,$

associativity : $x + (y + z) = (x + y) + z$

non – comm : $x \bullet (y) \neq y \bullet (x)$ for $x \neq y \neq \urcorner$

distributivity : $(x + y) \bullet z = (x \bullet z) + (y \bullet z).$

LoF calculus

Primary arithmetic

reflexivity : $\urcorner \bullet \urcorner = \varnothing$: **I2**

absorption : $\urcorner + \urcorner = \urcorner$: **I1**

associativity : $(\urcorner + \urcorner) + \urcorner = \urcorner + (\urcorner + \urcorner)$

distributivity : $(\urcorner + \urcorner) \bullet \urcorner = (\urcorner \bullet \urcorner) + (\urcorner \bullet \urcorner)$

$\urcorner \bullet (\urcorner + \urcorner) = (\urcorner \bullet \urcorner) + (\urcorner \bullet \urcorner)$

Primary algebra

absorption : $\overline{\overline{p}\rceil\ p}\rceil = \varnothing$: $\urcorner(\urcorner(p) + p) = \varnothing$: **J1**

commutativity : $p + q = q + p$

associativity : $p + (q + r) = (p + q) + r$

distributivity : $\overline{\overline{p}\rceil\ \overline{q}\rceil}\rceil\ r = \overline{\overline{p\ r}\rceil\ \overline{q\ r}\rceil}\rceil$: **J2**

$\urcorner(\urcorner(p) + \urcorner(q)) + r = \urcorner(\urcorner(p + r) + \urcorner(q + r)).$

Rules for substitution and equality $(=)$.

1.1.3. LoF as algebras

Further elaborations

There are other approaches to the study of the algebraic properties of the LoF that are more close to 2-element Boolean algebras. (Further elaborations are left to the reader.)

"In a more conventional notation an equational axiomatization can be written based on the reduction rules above:

(ab)c = a(bc),
a1 = a,
ab = ba,
(a'a)' = 1
((ac)'(bc)')' = (a'b')'c

This yields an equational algebra equivalent to a Boolean Algebra." (Mark Hopkins, 1996)
http://groups.google.com/group/sci.math.research

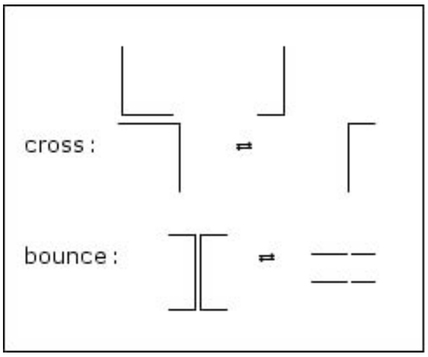
B. Banaschewski, On G. Spencer Brown's Laws of Form (1977)
http://projecteuclid.org/DPubS/Repository/1.0/

Philip Meguire, Boundary Algebra: A Simple Notation for Boolean Algebra and the Truth Functors (2007)
http://www.lawsofform.org/docs/Meguire_LoF.pdf

1.1.4. Generalizing approaches

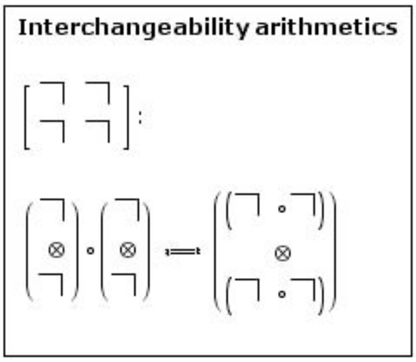
Idemposition
A(AB) = (AB)

"This calculus is based on the principle of idemposition *saying that: superposition of segments of the same color results in the cancellation of those segments"* (Kauffman, 2005, Map Reformulation)



"The Calculus of Idempositions is a Diagrammatic Language involving Closed Curves." (GSB)
The new property of "The calculus of Idempotence" is: *"Common Boundaries Cancel".*

Bifunctoriality
Another unifying approach for the *Laws of Form* was introduced as a *bifunctorial* interplay between the operations of concatenation and superposition: $\urcorner\!| \circ \urcorner\!| \rightleftarrows (\urcorner\,\urcorner) \otimes (\urcorner\,\urcorner)$.



How are *number* and *order* related in a bifunctoral approach?
The law of functorial interchangeability for the CI has to be set, it can't be deduced from the original mono-contextural Laws of Form. Interchangeability is introducing a new kind of *abstraction* beyond the isolated forms for *number* and *order*.

There is a reason to apply bifunctoriality to the CI because it contains two operators which are similar to *yuxtaposition* and *composition*, i.e. to serial and parallel application.
http://www.thinkartlab.com/pkl/media/Diamond%20Calculus/Diamond%20Calculus.html

1.1.5. Translation from CI to CL

1. Identity and order
CL identity: I(x) = x:
I2: *Order*: $\overline{\urcorner}\!| \leftrightharpoons \varnothing,\ \overline{a\urcorner}\!| \leftrightharpoons a,\ \overline{a\rceil a}\!| = \varnothing.$

The initial **I2** might therefore be considered as an operator (imperative), combinator or the name of an operator, hence as the mapping **I2**: $\overline{\varnothing\urcorner}\!|$ --> ∅ with **I2** as its operator. From a combinatory logical point of view such a functional operator of a morphism might be abstracted from its special realization, and set as an operator in the sense of a combinator. Hence, the mapping **I2**: $\overline{\varnothing\urcorner}\!|$ --> ∅ becomes the base of the CI-operator **I2**.

This interpretation of the Initials as operators instead of relations is supported explicitly by GSB himself with his wordings of *commands* and by his followers too.

I2: $\overline{\overline{\ }|}|$ --> ∅ :

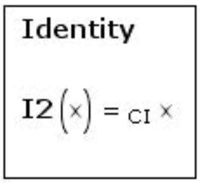
⏋ --> ⏋ ; ⏋ <--⏋ : ∅.

The double cross gets a notational definition as a *superposition* in a linearly written form:
⏋ (⏋) = $\overline{\overline{\ }|}|$. Thus, the brackets are indicating the superposition and are hindering to read the term as a concatenation: ⏋⏋.

In fact it is **I2**: $\overline{\overline{\emptyset|}|}$ --> ∅: **I**(∅) = ∅.

A more intuitive version of the Identity Initial is given by the general mapping **I**(a), i.e. **I2**: $\overline{\overline{a|}|}$ --> a, which also reads as: **I**(a): a --> $\overline{\overline{a|}|}$.

Therefore, the operational formulation of the identity initial, i.e. the Initial *Order* **I2**, reads as **I2**(x) = x.



Identity

$I2(x) =_{CI} x$

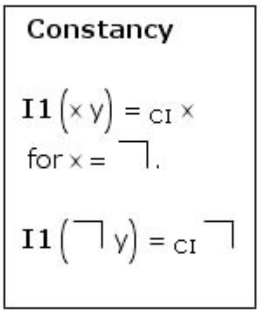
2. Constancy and number

CL-constancy: **K**(xy) = x:

Number: ⏋⏋ ⇔ ⏋ : **I1**
⏋⏋ --> ⏋,
⏋ --> ⏋⏋.

C3: ⏋*y* --> ⏋ , trivially: y = ⏋, y = ∅ => ⏋*y* --> ⏋ .
C"3: ⏋ --> ⏋*y*

C3 (x y) = x



Constancy

$I1(x\ y) =_{CI} x$
for x = ⏋.

$I1(⏋\ y) =_{CI}$ ⏋

Examples
(⏋*y*) $=_{CI}$ (y ⏋) $=_{CI}$ ⏋, but

I1 (⏋*y*) $!=_{CI}$ **I1** (y ⏋):

I1 (⏋*y*) $=_{CI}$ ⏋
I1 (y ⏋) = *y* = $\binom{\urcorner}{\emptyset}$: #

This correspondence with its conditions included, seems to be quite natural compaired to the identity function. The correspondence with the distributivity in CL follows without much interpretation.

3. Substitution and transposition

For CL :

Distributivity of substitution

S: $(S\,x\,y\,z) = (x\,z\,(y\,z))$.

For CI

Transposition

J2 : $\overline{\overline{p}\rceil\ \overline{q}\rceil}\rceil\, r = \overline{\overline{p\,r}\rceil\ \overline{q\,r}\rceil}\rceil$.

Hence, $(S\,p\,q\,r) = (p\,r\,(q\,r))$: $(S\ \overline{p}\rceil\ \overline{q}\rceil\ r) = \overline{p\,r}\rceil\ \overline{q\,r}\rceil$.

$\overline{\overline{p}\rceil\ \overline{q}\rceil}\rceil\, r \Longrightarrow \overline{\overline{p\,r}\rceil\ \overline{q\,r}\rceil}\rceil$.

Null

Substitution

$J2\,(x\,y\,z) =_{CI} (x\,z\,(y\,z))$

4. Principle of topological invariance

"Combinatory algebras (except the trivial one) are never commutative*, never* associative*, never* finite*, and never* recursive*."* (Barendregt)
http://mathgate.info/cebrown/notes/barendregt.php#7

Even if the translation of the CI-initials to CL-axioms is working, there is still no guarantee that the calculus as such is working at all. The reason is simple. There are crucial abstractions necessary to run the CI that are not manifestly implemented in the CI. Too much depends on intuition and informal instructions. One of the most crucial intuition of the CI is the "*topology invariance*" (Matzka/Varga) of its terms. This becomes clear if the graphematic base of the CI is considered and accepted.

"Remark. *Note that in working with the* primary algebra*, we take it for granted that elements of the algebra commute:*

A B = B A

for any algebraic expressions A and B. Certainly, we can observe that this is indeed an identity about the primary arithmetic*. It is just that we use this identity so frequently, that it is useful to take it as a given and not have to mention its use."* (Kauffman)

A dominant precondition of the definition of the CI is its topological invariance, i.e. commutativity in respect of the *positions* of its terms in an arrangement. This is in strict contrast to the basic non-commutativity of terms in the CL. For the CI, commutativity of terms is an undefined prerequisite of the very calculus based on convention. Because of this abstraction, problems with the formalization of a proper calculus are following automatically.

Summary
This notion of commutativity as Kauffman's remark shows is still very vague. As a summery of the exercise some results might be collected.
Commutativity for the CI holds for *concatenation* of arithmetic and algebraic expressions. Commutativity doesn't hold, trivially, for superposition, except for the case of notational identity of operator and operand of the superposition, say $\overline{\ \overline{\ }\rceil_1}\rceil_2 = \overline{\ \overline{\ }\rceil_2}\rceil_1$ with $\overline{\ }\rceil_1 = \overline{\ }\rceil_2$.

This confirms, again, the observation of Varga/Matzka that the CI-commutativity is not just an *algebraic* property as in the sense of Kauffman et al but a *semiotic* or, as it was emphasized in other papers, a *graphematic* property.

"*A third deviation from classical semiotics is less obvious: the* commutativity *of the concatenation operation. For any two terms "a" and "b" the terms "ab" and "ba" are identical. That this is indeed a semiotic identity (and not just a logical equality) has been stressed by Varga.*" (Matzka, 1993)

Nevertheless, this insight into the *semiotic* deviation, in respect of the type/token-relation, for semiotic concatenation, in contrast to "encloser" (superposition), is still not very clear. That is, taken at face value, it boils too quickly down to the classical algebraic situation of commutativity.

The proposed solution, presented in my previous papers, stresses the fact that the CI-semiotics and its syntactical structure is not a tree, based on *mark* and *unmark*, cross and blank, but is, from the very beginning of the calculus, a commutative structure. This structure makes no sense for a 2-element CI-algebra with {cross, blank} but needs a "pattern"-oriented approach of at least 3 different configurations: (aa), (ab), (bb), with (ab) $=_{CI}$ (ba).

On this graphematic level, the famous "commutativity" holds in a strictly deviant sense at least in respect of an identity-based semiotics. And that just blocks any reduction of the CI to a Boolean algebra. Usually, the concatenation of terms in the CI gets a logical interpretation as *conjunction* or as *disjunction*. Obviously, both are commutative: $p \wedge q = q \wedge p$. It seems then, that on this level of reflection, commutativity of concatenation appears as not such special as pronounced.

The presented reflections and the following arguments may give some insights into the different levels of "commutativity" in the CI. But it is still not specially clear, if those insights still remain in the conceptual and technical framework of the CI. If not, it seems difficult to "save" the CI from being reduced to a conventional calculus based on 2 atomic states: cross and blank.

Without the insistence on the *"topology-invariant"* features, it seems that the concept and some primary formalization of the CI is not just more or less "isomophic" to propositional logic but also to Combinatory Logic, and with that to the Lambda Calculus too.

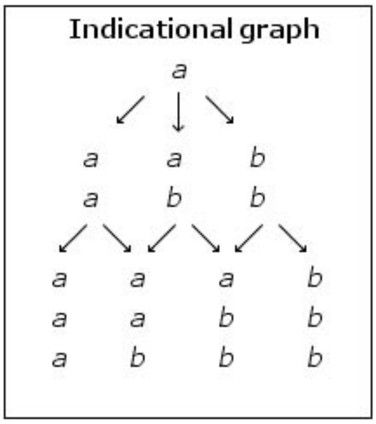
Conventional reception of the CI was mainly interested in the re-entry form and its speculations, and not much into the epistemological and technical aspects of the basic calculus of indication as such.

Combinatorics

CI – graphematics : $a \neq b,\ (aa) \neq (bb),\ (ab) = (ba).$

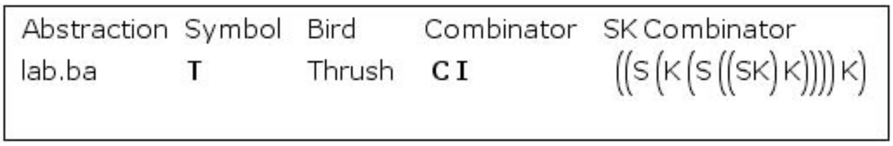
$\text{Brown}_{(n,\, m)} = \binom{n+m-1}{n}.$

Catalan number of brackets for $\forall n \in N$: $C_n = \frac{1}{n-1} \binom{2n}{n}$.



Commutativity and the combinator T

Some further "elaborations" of the commutativity topic is accessible with the help of the combinator **T.**



Abstraction	Symbol	Bird	Combinator	SK Combinator
lab.ba	T	Thrush	**C I**	((S(K(S((SK)K))))K)

http://www.angelfire.com/tx4/cus/combinator/birds.html

$$\boxed{\mathbf{T}\,a\,b =_{CL} b\,a}$$

CI

$$\boxed{\mathbf{T}\left(p\,q\right) =_{CI} \left(p\,q\right)}$$

$$\begin{pmatrix} \mathbf{T}\,a\,b =_{CI} ab \\ \mathbf{K}\left(a\,b\right) = a \\ \mathbf{K}''\left(a\,b\right) = b \end{pmatrix} \Longrightarrow \mathbf{K}\,a\,b =_{CL} \mathbf{K}''\left(\mathbf{T}\left(ab\right)\right);\ \text{but}\,\mathbf{K}\,a\,b \neq_{CI} \mathbf{K}\left(\mathbf{T}\left(ab\right)\right).$$

A solution of the conflict is offered by the rules of *standard normal* forms. Also the operator **T** has an invariant result in the CI, its abstraction, i.e. **T**ab = ab, has to be followed by the choice of a standard normal form presentation of the terms “ab”. Hence, also **T**ab = ba, and ab $=_{CI}$ ba, the representation of the abstraction **T** demands an invariant representation in the calculus. This is nothing new but has to be considered again.

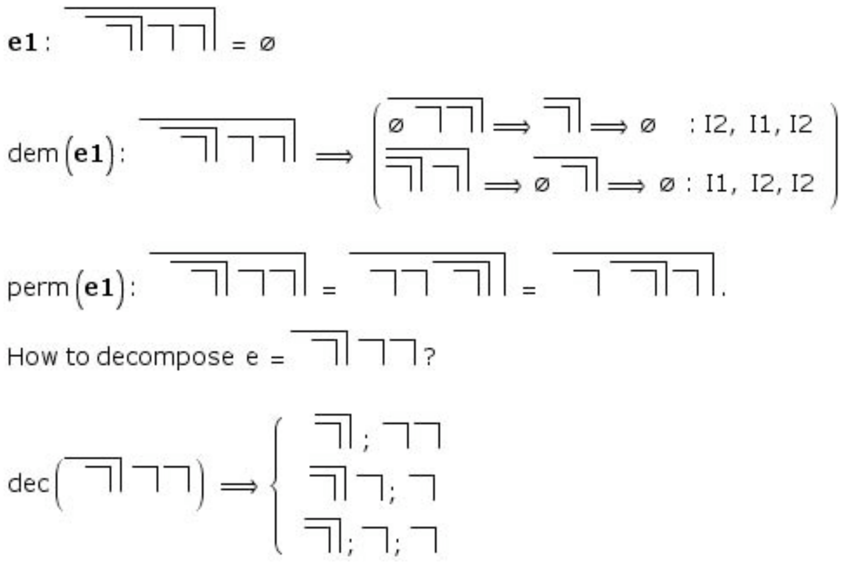
Therefore, in a isolated situation, the formula (ab) $=_{CI}$(ba), i.e. for a, b ∈ CON (concatenation), may hold but this is not the case if the terms are involved with an operation and are including variables, say, albeit
(ab) $=_{CI}$(ba), **K**ab !$=_{CI}$ **K**ba.

Definition of a CI-expression

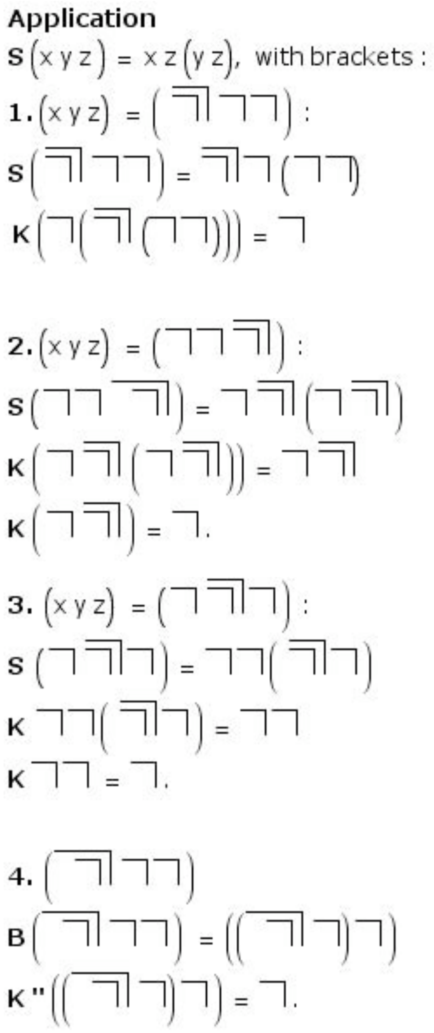
Hence, a CI-expression *e* is defined as an abstraction over **T**:
e ∈ CI iff ∀ a, b ∈ CI: (a b) = e_1, **T**(a b) = e_2: $e_1 \cong_{CI} e_2$.

e ∈ CI iff e $\cong_{CI}$ **T**(e) .

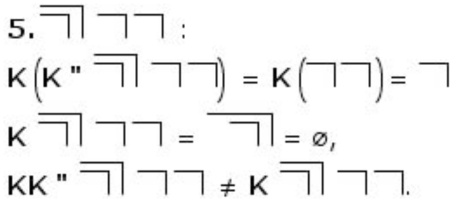
1.1.6. Demonstrations



Instead of brackets and their use as superposition marker the semicolon ";" is used to avoid confusion.



short:
Brackets for notational order and brackets as superposition markers.



Result of 5.
There is a conflict between the rules for **T** and **K** applied to the CI.

1.2. Church-Rosser confluence for CI expressions

1.2.1. Church-Rosser for CL

CL-expressions are well defined with variable x, primitive functions P, and applications (E_1E_2) of combinatory terms E_1 and E_2. Brackets are associative: (E1 E2 E3 ... En) = (...((E1 E2) E3)... En).

Associativity of applications is not to confuse with the permutativity of terms in the CI.
Hence,

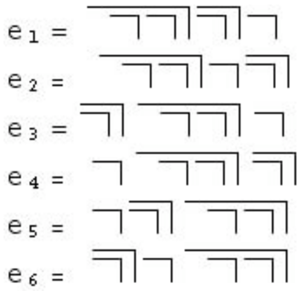
$$E_1, E_2 \in CL : (E_1\ E_2)\ !=_{CL}\ (E_2\ E_1).$$

Combinatory Logics are well documented.

http://maths.swan.ac.uk/staff/jrh/papers/JRHHislamWeb.pdf
Church-Rosser for morphogrammatics are sketched at:
http://memristors.memristics.com/Church-Rosser%20Morphogrammatics/Church-Rosser%20in%20Morphogrammatics.pdf

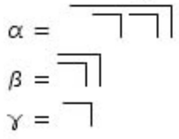
1.2.2. Church-Rosser for CI

A simple CI-example for a CI-expression *e* is given by the fact of the permutation-invariance of its marks. A CI-expression *e* is an abstraction of its possible realizations based on the set of its terms and the position of the terms in the expression. The implicit algebraic axiom of the commutativity, ab = ba, of the concatenation of terms is ruling the game.



$\forall i, j : e_i =_{CI} e_j,\ i, j = 1, \ldots, 6$

$e = \text{perm}\{\alpha, \beta, \gamma\}$

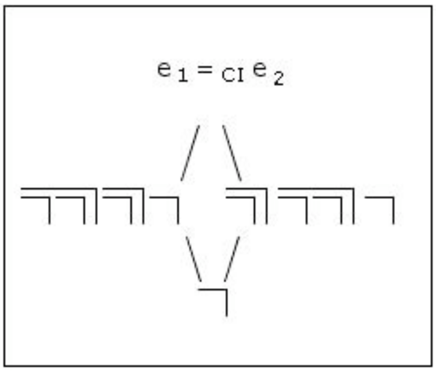


$\text{type} = \{\alpha, \beta, \gamma\}$

Hence, a CI-expression or arrangements consists of the *marks* {⌝, ∅}, the *combinations* and the *positions* of the combinations. Combinations are closed under the dominating mark. The concept of positions of terms of an expression is not used in the CI. It seems to be one of the hidden abstractions.

expr/type	pos1	pos2	pos3
e_1	α	β	γ
e_2	α	γ	β
e_3	β	α	γ
e_4	γ	α	β
e_5	γ	β	α
e_6	β	γ	α

Therefore, a CI-expression is an *abstraction* of the realizations of the permutation of its terms. What is considered commonly as a CI-expression is a *representation* or a standard normal form of an expression and not the concrete expression defined by the explicit and hidden rules of the CI. This observation holds under any "umbrella" of a dominating form, too.



Thus a value for any arrangement can be supposed if the arrangement can be simplified. But it is plain that some arrangements can be simplified in more than one way, and it is conceivable that others might not simplify at all. (GSB, LoF, p. 10)

Also the CI is governed by the hidden law of permutational invariance, a reduction of a CI-expression delivers, at least in the finite case, an identifiable value. All CI-expressions are reducible to the value *mark* or *unmark*. Because the CI is defined by an "elementaristic" approach, two marks are unambiguously equal or non equal. Therefore, the CI belongs nevertheless to the class of identity calculi.

A pattern-oriented approach is able to keep the property of permutational invariance in the result of a demonstration. Hence, a result of a demonstration or reduction might differ in the order of its values, like (⌉ ∅) $=_{\text{CI-pat}}$ (∅ ⌉).

"The simplification of an expression is unique." (LoF, p. 14)

Permutation-invariance disappears in the results of a CI demonstration or calculation.

1.3. Reductions of axioms

1.3.1. Reduction of basic operators

To define a similarity or even an isomorphism between two calculi is a straightforward exercise. But to find a way to apply the translation and to find similar reductions of axioms, is a more challenging entertainment.

For CL

The reduction of the identity combinator **I** of the CL is standard. The identity **I**(x) might be defined by the combinators **S** and **K** only.

$$\mathbf{I}(x) = x :$$
$$((\mathbf{SKK})x) \longrightarrow (\mathbf{SKK}x) \longrightarrow (\mathbf{K}x(\mathbf{K}x)) \longrightarrow (\mathbf{I}x).$$

(**K**x(**K**x)) is a superposition of **K**x onto **K**x, and not a concatenation. Hence, (**K**x(**K**x)) is an application of the combinator **K** as an operator on the combinator **K** as an operand.

This reduction is applied to the CI :

$$((\mathbf{J2\,I1\,I1})x) \longrightarrow (\mathbf{J2\,I1\,I1}\,x) \longrightarrow (\mathbf{I1}\,x(\mathbf{I1}\,x) \longrightarrow x = (\mathbf{I2}\,x),$$

With $x = \urcorner\urcorner$:

$$(\mathbf{I1}\urcorner\urcorner)(\mathbf{I1}\urcorner\urcorner) = (\mathbf{I1}\urcorner\urcorner)(\urcorner) = \urcorner(\urcorner) = \overline{\urcorner}\rceil.$$

$\mathbf{I1}x(\mathbf{I1}x) = (\mathbf{I}x) : \mathbf{I2}x.$ Distr = **J2**, **I1** = **K**, **I** x by definition : $\overline{\urcorner}\rceil$.

Therefore, it can be shown by means of CL, that the initial **I2** can be eliminated without loss in the CI, resulting in the new calculus CI'= (I1, J2, app), similarily to the reduction of the identity combinator **I**(x) in CL. Thus*, application* in CL is *superposition* in CI, *concatenation* in CL is concatenation in CI.

Reduction in CI

$\mathbf{I2}\,x \Leftarrow (\mathbf{J2\,I1I1})\,x$
for any x

In Brownian terms

Order x $\Leftarrow$ (Transposition Number Number) x
for any x

Position J1

$$\overline{\overline{p}\rceil\, p}\rceil = \varnothing \quad :: \quad \overline{\begin{matrix}\overline{p}\rceil & \\ & p\end{matrix}}\Bigg\rceil = \varnothing : \overline{\begin{matrix}\overline{\urcorner}\rceil & \\ & \urcorner\end{matrix}}\Bigg\rceil = \overline{\begin{matrix}\varnothing & \\ & \urcorner\end{matrix}}\Bigg\rceil = \overline{\urcorner}\rceil\ ,\ \overline{\begin{matrix}\overline{\varnothing}\rceil & \\ & \varnothing\end{matrix}}\Bigg\rceil = \overline{\urcorner}\rceil = \varnothing.$$

$$\overline{x}\rceil\,(\overline{x}\rceil) = \overline{\overline{x}\rceil x}\rceil : (\mathbf{S}\urcorner(\urcorner))\,x = (\urcorner x(\urcorner x)) = \overline{\overline{x}\rceil x}\rceil$$

$$\overline{p}\rceil\,(\overline{p}\rceil) = \overline{\overline{p}\rceil p}\rceil : (\mathbf{S}\urcorner(\urcorner))\,p = \urcorner p(\urcorner p) = \overline{\overline{p}\rceil p}\rceil.$$

Position J1

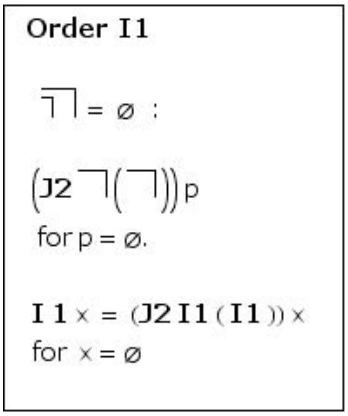
$\overline{\overline{p}\rceil\, p}\rceil = \varnothing$:

$(\mathbf{J2}\urcorner(\urcorner))\,p = \varnothing$
for any p

Position J1

$\overline{\overline{p}\rceil\, p}\rceil = \varnothing$:

$\mathbf{J1}\,x = (\mathbf{J2\,I1}(\mathbf{I1}))\,x$
for any x



Order I1

$\overline{\ }\rvert\,\overline{\ \ }\rvert = \varnothing$:

$\left(J2\ \overline{\ \ }\rvert\left(\overline{\ \ }\rvert\right)\right)p$

for $p = \varnothing$.

$I1\,x = (J2\,I1\,(I1))\,x$

for $x = \varnothing$

Results

The Initials **I2** for *Order* of the primariy arithmetic and **J1** for *Position* of the primary algebra are reducible to **I1** for *Number* of the primary arithmetic and **J2** for *Transposition* of the primary algebra**.**

Order and Position are definable by the equivalents of the CL combinators **K** and **S**.

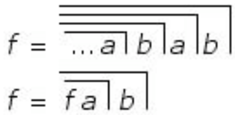
That is to say, the CI Initials for *Order* and *Position* are *redundant*.

The proofs are accepting a strategic mix of primary *arithmetic* and primary *algebra*, i.e. **I1** and **J2.**

1.3.2. Definition of self-referential operators

Re-entry operator

Self-reference in the CI is build by an *adhoc* construction of abstraction but not by the application of CI rules.



$$f = \overline{\overline{\overline{\overline{\ldots a}\rvert\, b}\rvert\, a}\rvert\, b}\rvert$$

$$f = \overline{\overline{f\,a}\rvert\, b}\rvert$$

And so on. And by *meditation* you get: $f = \overline{f}\rvert \equiv \underline{f}\rvert$, LoF' s final script for self – reference (Varela)

But this is just part of magic and has no foundation in the *appartus* of the CI. A similar figure of introduction occurs by *meditation* on iterativity with Heinz von Foersters *Eigenvalues* of his Uroboros. In contrast, Combinatory Logic is able to *construct* its beasts. In this sense, the CL is "deeper than truth", simply because these CL-contradiction-free constructions are producing contradictions in the context of a *logical* interpretation.

http://www.thinkartlab.com/pkl/lola/poly-Lambda_Calculus.pdf

Self-reference CL

```
--> (S I I (S I I))
|         ↓
|    (I (S I I (I (S I I)))
|         ↓
|    (S I I (I (S I I)))
|         ↓
<--  (S I I (S I I))
```

Y-operator in CL

Y $=_{CL}$ f (**Y**f).

One of the many classical definitions of **Y**:

Y f **= (W(B** f**)) (W(B** f**)) = WS (BWB)** f

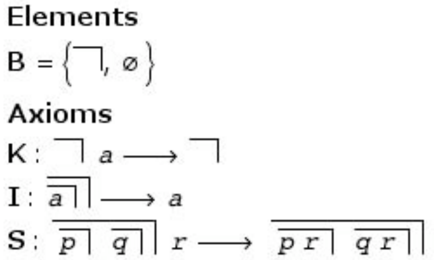
with
W x y = x y y : **W** = **SS** (**SK**)
B x y z = x (y z) : **B** = **S** (**KS**) **K**

Proof of **Y**: **Y** f = f (**Y** f):
Yf = **WS** (**BWB**) f
= **BWB** f (**BWB** f)
= **W** (**B** f) **BWB** f)
= f (**Y** f).

Self-reference in operational CI
The operational version of the CI might mimicking the CL construction of a CI-**Y** operator.

```
--> (J2 I2 I2 (J2 I2 I2))
|          ↓
|    (I2 (J2 I2 I2) (I2 (J2 I2 I2)))
|          ↓
|    (J2 I2 I2 (I2 (J2 I2 I2)))
|          ↓
<--  (J2 I2 I2 (J2 I2 I2))
```

1.4. An alternative calculus out of the form



Elements
$B = \left\{ \overline{\quad}|, \varnothing \right\}$
Axioms
K: $\overline{\quad}| \; a \longrightarrow \overline{\quad}|$
I: $\overline{\overline{a}|}| \longrightarrow a$
S: $\overline{\overline{p}| \; \overline{q}|}| \; r \longrightarrow \overline{\overline{p\,r}| \; \overline{q\,r}|}|$

2. Pattern-oriented formalization

2.1. Orientation

Motivations
Also it seems to be difficult to characterize the CI properly in its own conception, because of a serious lack of original elaborations of the calculus by George Spencer-Brown, my own main interest into this confusing situation is to find genuine “formal systems” that correspond to the scriptural levels of the proposed graphematics.

The identity level of the graphematic architecture is obviously perfectly covered by standard mathematical formal systems, logics and formal languages.
The purely non-identity, i.e. kenogrammatic levels, are principally well accessible by the conceptions and apparatus of morphogrammatics.

The formal system, covered or hidden by the CI has similarities to the *mixed* language of kenogrammatics and identity system of “topology” invariance. I am not aware of another approach to deal with this graphematic level of mixed formal languages as with a generalized approach to the Brownian “*indicational*” systems.

The motivation to the complementary system to the generalized indicational system, proposed by the generalized calculus of *differentiation*, has the function to offer a formal interpretation to the Mersenne languages as the second mixed language of graphematics.

Leaving the battlefield

The easiest way to find peace with the Brownian ambitions and its defenders is to find a compromise in the valuation of the whole drama. It seems, as many have shown before, that the calculus of indication, as it is developed by Spencer-Brown, is *not* delivering the proclaimed revolution in formal thinking. The apparatus, as it exists and as far as it works, is not much more than a deviant notation for a well known conceptual structure.

The hope to find a structure or even a realm "*deeper than truth*" (Varela) uncovered by the CI has not been fulfilled. The intention may fit well into the project of graphematics with its different types of scriptural systems. But the Brownians are not aware of graphematical systems as they have been introduced since 1962 by Gotthard Gunther.

What makes the difference concerning the claims of GSB in respect of his calculus are the non-formal declarations. One aspect is leading to the underlying commutative graph-structure of the concept of the CI. Opting for this approach is motivating an indicational calculus of patterns, instead of atomic elements of arbitrary complexity.

The Brownian community, still alive, especially in Germany, seems still not to be aware of the many different other approaches to formalization beyond classical logic, notational inventions and profound philosophical interventions concerning the well known structures, albeit from different angles, perspectives, motivations and practical considerations. All developed in the long history after George Boole, Gottlob Frege, C.S. Peirce and many others, and up to date in mathematical linguistics, programming languages and computer science too.

There is quite clearly also no special merits to earn with GSB's concept of *re-entry* for tackling self-referential formal structures. A lot of work has been done from the side of the followers of GSB in this direction but mainly ignoring the classical results of recursive function theory, modal logics and other formalizations of self-reference and the strange fact, that there is no proper connection between the CI and self-referentiality established.

2.1.1. Sketch of a morphogrammatic modeling

Tree-structure of Combinatory Logic

CL

Syntax

(0) terms = {K, S, I, T}, variables = {x, y, z,...}, operator *
(i) all variables and constants are CL-terms,
(ii) if X and Y are CL-terms, then (X Y) is a CL term.

Semantics

K: LA-tree(K(xy)) => x
S: LA-tree(Sxyz) => 2-tree(2-tree(xz), 2-tree(yz))
T: LA-tree(Txy) => 2-tree(y x)
I: tree(x) => tree(x).

Algebra

CL = (CL, •, S, K)
Axioms and Rules

Reductions:
CL axioms are not equalities, "⇔", like in the CI but reduction rules with "-->".

It seems that there is no chance to build a primary "*topology-invariant*" structure out of the tree-rules for the CL. Again, **T**ab !=$_{CL}$ ab.

Tree-structure for the calculus of indication

CI

Syntax

(0) terms = {⌉, ⌀}, variables = {p, q r, s,...}, operators = {*, ^},
(i) all variables and constants are CI-terms,
(ii) if X and Y are CI-terms, then (X * Y) and (⌉ ^(Y)) is a CI term.

Semantics

I1 : $\urcorner\urcorner \Longleftrightarrow \urcorner$

I2 : $\overline{\urcorner}| \Longleftrightarrow \varnothing$

Rules

J1 : $\overline{\overline{p}| \; p}| \Longleftrightarrow \varnothing$

J2 : $\overline{\overline{p}| \; \overline{q}|}| \; r \Longleftrightarrow \overline{\overline{p\,r}| \; \overline{q\,r}|}|.$

Graph – structure of CI – com

As an improvised solution :

J2 – com : $\text{LA} - \text{tree}(\mathbf{J2}\, p\, q\, r) \Longrightarrow 2 - \text{tree}(2 - \text{tree}(p\, r),\ 2 - \text{tree}(q\, r)) \wedge q \cong_{CI} r.$

$\forall x, y, z \in \text{CI}:\ Z(xyz) \text{ iff } \mathbf{S}(xyz) \wedge \mathbf{T}xy =_{CI} xy.$

Or simply as the hidden assumption for the commutativity of concatenation **J0** :

J0 : $\forall x, y \in \text{CI}:\ x\,\hat{}\,y =_{CI} y\,\hat{}\,x.$

In CL – terms :

J0 : $\forall x, y \in \text{CI}: \mathbf{T}xy =_{CI} xy.$

$\mathbf{CI} = (\{\urcorner, \varnothing\}, \text{I1}, \text{I2}, \text{J0}, \text{J1}, \text{J2}, *, \hat{}, \Longleftrightarrow)$

It seems that there is no chance to build a primary "*topology-invariant*" structure out of the tree-rules for the elementaristic CI.

Unfortunately, the *intentions* of the CI suggest a commutative graph instead of a binary tree.

In contrary, again,
"*Every finite expression has a unique simplification.*" (T3 in LoF);

Because the CI has in fact a base of two elements, {⌉, ⌀}, and not of just one element {⌉} as it is said, a mapping of the 2-element base onto the syntax produces a binary tree. On this tree, configurations (expressions) are obviously defined unambiguously by their path in the tree. The path is readable in both directions, and therefore, rules are a kind of equalities, with "⇔", and not strictly uni-directional reductions like in CL with "-->".

Hence,
"Any pa *(or sentential logic) formula B can be viewed as an* ordered tree *with branches."*

This comfortable, but annoying situation, disappears, if the declared *intentions* and *ambitions* of the Brownian declarations are taken by face value.

Because of the commutativity of its basic terms, as it is understood by the declared ambitions, the simple syntactic and semantic, or indicational hierarchy, has to be replaced by a commutative structure, where configurations are determined in different ways, as results of different paths in the graph.

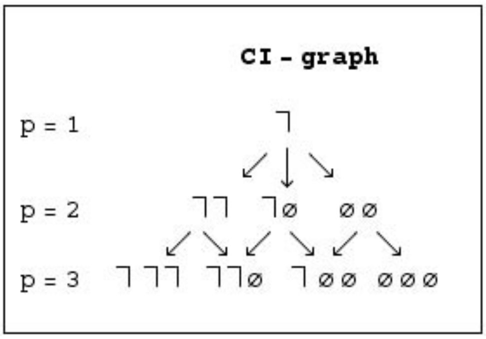
Therefore, a CI configuration in a pattern-oriented setting, will be defined by several paths in the graph.

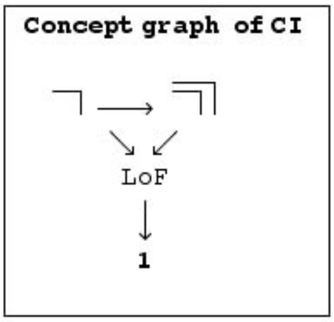
$$\textit{Example}: \urcorner\urcorner\varnothing \cong \begin{pmatrix} x \longrightarrow \urcorner\urcorner \longrightarrow \urcorner\urcorner\varnothing \\ x \longrightarrow \urcorner\varnothing \longrightarrow \urcorner\urcorner\varnothing \end{pmatrix}.$$

2.1.2. Contextural modeling of patterns

A slightly less annoying modeling might be opened up by a *contextural* approach that is disseminating the classical 2-state calculus of indication over different contextural loci. A 3-contextural modeling is disseminating a 2-state CI over 3 different loci of a reflectional configuration. (cf. Diamond Calculus)

The contextural approach is decomposing patterns into elementary parts, while the morphogrammatic approach to patterns is operating on their morphogrammatic laws.





$$\textbf{Brown}(2,2): \{\urcorner, \varnothing\}^{1}_{/\mathrm{CI}} \longrightarrow \{\urcorner, \varnothing\}: \binom{n+m-1}{n} : \binom{2+2-1}{2} = 3$$

$\urcorner A$	A	$\overline{\urcorner A}\vert$
$\urcorner$	$\urcorner$	$\varnothing$
$\urcorner$	$\varnothing$	$\varnothing$

.

Permutation versus braiding for double – crossing

Like double – negation, double – crossing results in the initial state :

$\mathbf{CI}: \overline{\overline{\varnothing}|}| \Longleftrightarrow \varnothing :$

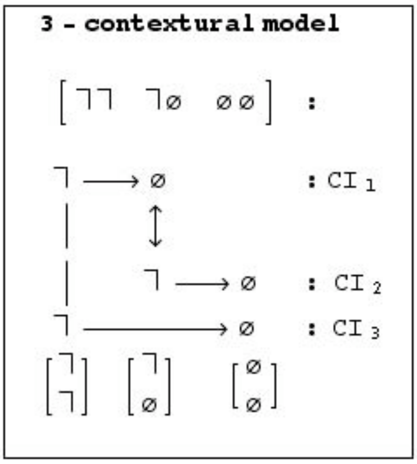
$\overline{\overline{\varnothing}|}| \to \varnothing : \overline{\overline{\varnothing}|}| \longrightarrow \overline{\varnothing}| \longrightarrow \varnothing,$

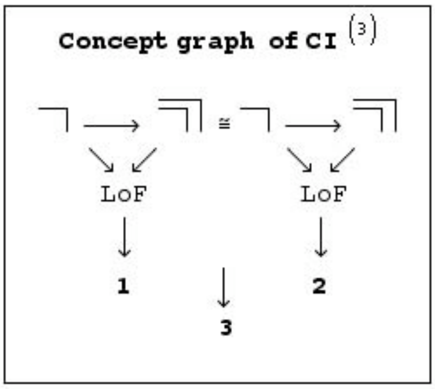
$\varnothing \to \overline{\overline{\varnothing}|}| : \varnothing \longrightarrow \overline{\varnothing}| \longrightarrow \overline{\overline{\varnothing}|}|.$

A pattern-related approach is involved into topological chances. Instead of a permutative inversion dictated by an identity space of a set-theoretic permutation, a *braiding* in a topological space is defined that is taken into account the differences between the directions of the construction and destruction of the superposition "$\overline{\overline{\varnothing}|}|$":

$$\textbf{Braided crossing}: \overline{\overline{\varnothing}|}| : \overline{\overline{\varnothing}|_1}|_2 \neq \overline{\overline{\varnothing}|_2}|_1 \Longrightarrow \overline{\overline{\overline{\overline{\varnothing}|_1}|_2}|_2}|_1 = \overline{\overline{\overline{\overline{\varnothing}|_2}|_1}|_1}|_2.$$

2.1.3. Dissemination of CIs





Order

$\urcorner \rightarrow \emptyset \ : \forall i : \urcorner_i > \emptyset_i \wedge i = 1 < 2 < 3$

Algebra

$\mathbf{CI} = (\{\urcorner, \emptyset\}, \mathrm{I1}, \mathrm{I2}, \mathrm{J1}, \mathrm{J2}, \mathrm{conc}, \mathrm{sup}, =)$

$\varsigma : \{\mathrm{I1}, \mathrm{I2}, \mathrm{J1}, \mathrm{J2}\} \times \{\urcorner, \emptyset\} \longrightarrow \{\urcorner, \emptyset\}$

$\varsigma : \{\mathrm{ops}\} \times \{\urcorner, \emptyset\} \longrightarrow \{\urcorner, \emptyset\}$

$\mathrm{ops} : \{\urcorner, \emptyset\} \longrightarrow \{\urcorner, \emptyset\}$

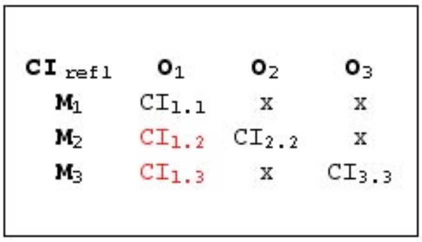
$\mathbf{CI} = (\mathrm{ops}, \{\urcorner, \emptyset\}) = (\mathrm{ops}, \mathrm{elements})$

Strictly reflectional constellation matrix CI $^{(3,1)}$

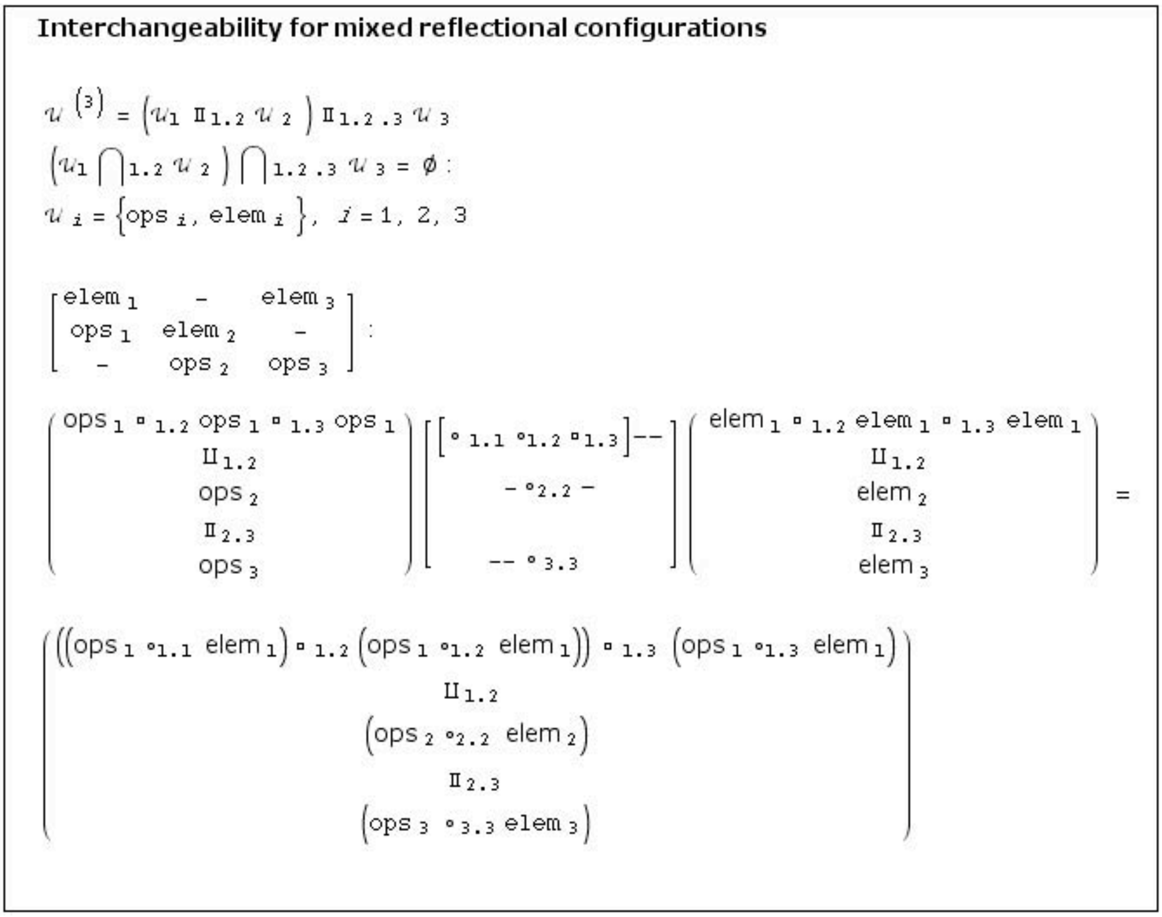
$\mathrm{CI}^{(3,1)}$	O_1	O_2	O_3
M_1	$\mathrm{CI}_{1.1}$	-	-
M_2	$\mathrm{CI}_{1.2}$	-	-
M_3	$\mathrm{CI}_{1.3}$	-	-

Null

Mixed reflectional constellation



$\mathbf{CI}_{\mathrm{refl}}$	$\mathbf{O}_1$	$\mathbf{O}_2$	$\mathbf{O}_3$
$\mathbf{M}_1$	$\mathrm{CI}_{1.1}$	x	x
$\mathbf{M}_2$	$\mathrm{CI}_{1.2}$	$\mathrm{CI}_{2.2}$	x
$\mathbf{M}_3$	$\mathrm{CI}_{1.3}$	x	$\mathrm{CI}_{3.3}$



Interchangeability for mixed reflectional configurations

$$\mathcal{U}^{(3)} = \left(\mathcal{U}_1 \amalg_{1.2} \mathcal{U}_2\right) \amalg_{1.2.3} \mathcal{U}_3$$

$$\left(\mathcal{U}_1 \bigcap_{1.2} \mathcal{U}_2\right) \bigcap_{1.2.3} \mathcal{U}_3 = \emptyset :$$

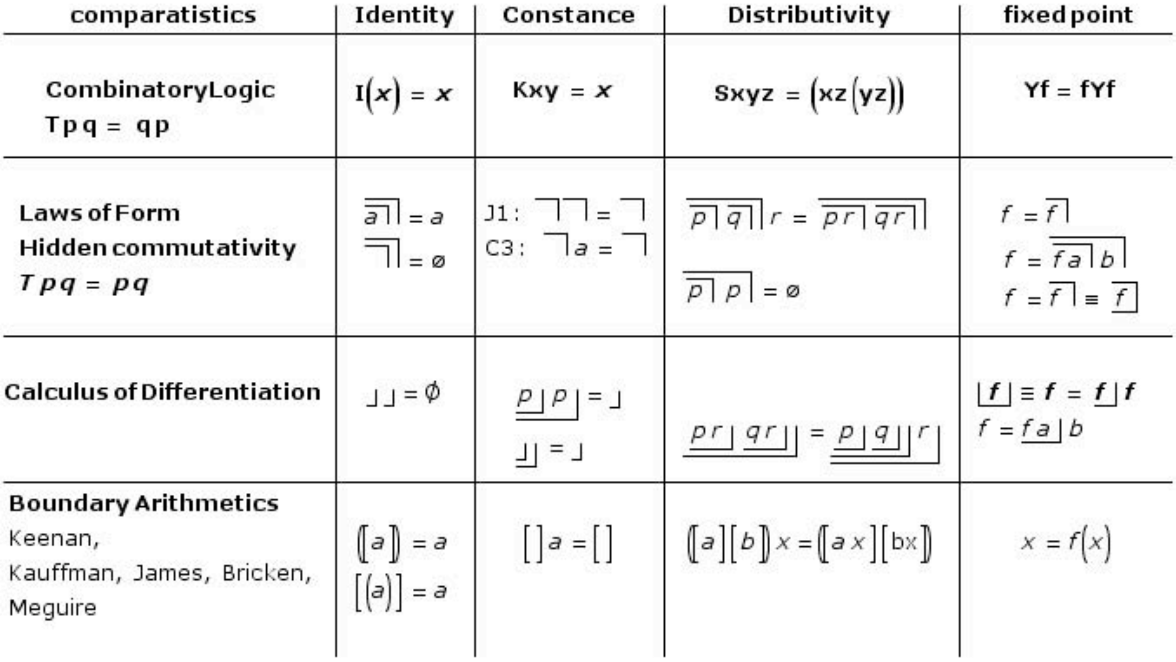
$$\mathcal{U}_i = \{ops_i, elem_i\}, \quad i = 1, 2, 3$$

$$\begin{bmatrix} elem_1 & - & elem_3 \\ ops_1 & elem_2 & - \\ - & ops_2 & ops_3 \end{bmatrix} :$$

$$\begin{pmatrix} ops_1 \,\square_{1.2}\, ops_1 \,\square_{1.3}\, ops_1 \\ \amalg_{1.2} \\ ops_2 \\ \amalg_{2.3} \\ ops_3 \end{pmatrix} \begin{bmatrix} [\circ_{1.1}\ \circ_{1.2}\ \square_{1.3}]-- \\ -\ \circ_{2.2}\ - \\ \\ --\ \circ_{3.3} \end{bmatrix} \begin{pmatrix} elem_1 \,\square_{1.2}\, elem_1 \,\square_{1.3}\, elem_1 \\ \amalg_{1.2} \\ elem_2 \\ \amalg_{2.3} \\ elem_3 \end{pmatrix} =$$

$$\begin{pmatrix} \left(\left(ops_1 \circ_{1.1} elem_1\right) \square_{1.2} \left(ops_1 \circ_{1.2} elem_1\right)\right) \square_{1.3} \left(ops_1 \circ_{1.3} elem_1\right) \\ \amalg_{1.2} \\ \left(ops_2 \circ_{2.2} elem_2\right) \\ \amalg_{2.3} \\ \left(ops_3 \circ_{3.3} elem_3\right) \end{pmatrix}$$

3. Comparatistics Table



comparatistics	Identity	Constance	Distributivity	fixed point
CombinatoryLogic **$Tpq = qp$**	$\mathbf{I}(x) = x$	$\mathbf{Kxy} = x$	$\mathbf{Sxyz} = (\mathbf{xz}(\mathbf{yz}))$	$\mathbf{Yf} = \mathbf{fYf}$
Laws of Form **Hidden commutativity** **$Tpq = pq$**	$\overline{\overline{a\vert}\vert} = a$ $\overline{\overline{\ \vert}\vert} = \emptyset$	J1: $\overline{\ \vert}\,\overline{\ \vert} = \overline{\ \vert}$ C3: $\overline{\ \vert}\,a = \overline{\ \vert}$	$\overline{\overline{p\vert}\,\overline{q\vert}\vert}\,r = \overline{\overline{pr\vert}\,\overline{qr\vert}\vert}$ $\overline{\overline{p\vert}\,p\vert} = \emptyset$	$f = \overline{f\vert}$ $f = \overline{\overline{fa\vert}\,b\vert}$ $f = \overline{f\vert} \equiv \underline{f\vert}$
Calculus of Differentiation	$\underline{\ \vert}\,\underline{\ \vert} = \emptyset$	$\underline{\underline{p\vert}\,p\vert} = \underline{\ \vert}$ $\underline{\underline{\ \vert}\vert} = \underline{\ \vert}$	$\underline{\underline{pr\vert}\,\underline{qr\vert}\vert} = \underline{\underline{\underline{p\vert}\,\underline{q\vert}\vert}\,r\vert}$	$\underline{\vert\mathbf{f}\vert} \equiv \mathbf{f} = \underline{\mathbf{f}\vert}\,\mathbf{f}$ $f = \underline{fa\vert}\,b$
Boundary Arithmetics Keenan, Kauffman, James, Bricken, Meguire	$([a]) = a$ $[(a)] = a$	$[\,]a = [\,]$	$([a][b])x = ([a\,x][\mathrm{bx}])$	$x = f(x)$

http://www.mathematica-journal.com/issue/v5i4/columns/maeder/35-41roman.54.mj.pdf