DISTRIBUTION IS NOT AXIOMATIC

William Bricken
October 1986

DISTRIBUTION is not a necessary axiom for Boolean calculus.

This means that the entire basis of the mathematics is built upon *erasure* and *creation* actions. No concept of rearrangement is needed, therefore there is no representational explosion caused by rearrangement.

This has profound implications for the interpretation for developmental psychology, and for cognitive modeling. What we have is an extremely principled basis to describe the origin of thought, based on *constructive* activity.

I'll show you the formalism, and we can bliss out about what can be seen and explained later. An outline of the exposition from first principles:

ARITHMETIC

( ) ==> ( ) ( ) ==> ( ) (( ))

Initialization

We represent *ourselves*, creating the *initial distinction*,

=/=> ( )

Distinction

We use ( ) to provide two worlds. The outer is *pervasive*; the inner is *empty*.

Expansion

We populate the outer world with a *replicate*,

( ) ==> ( ) ( )

by replicating the initial distinction.

We populate the inner world with a replicate

( ) =/=> (( ))

by replicating the initial distinction on the inside.

We can also populate the inner world with the replicate

( ) ( ) ==> ( ) (( ))

by relying on the pervasiveness of the replicate in the outer world.

Contraction

We recognize that the replicate is redundant

( ) ( ) ==> ( )

We recognize that the inner replication revokes the initial distinction

(( )) ==>

by the process

=/=> ( ) =/=> (( )) ==>

Completion

We accept the arithmetic.

ALTERNATIVE ARITHMETIC

==> (( )) ==> (( ) ( ))

We recognize *ourselves*, creating the *initial distinction*,

==> (( ))

We use (( )) to identify two worlds. The outer is imaginary and empty; the inner, populated.

We create a replicate

(( )) ==> (( ) ( ))

which is redundant....

ALGEBRA

Generality

We generalize our ability to populate the outer world

( ) ==> ( ) A   DOMINION

We generalize our ability to populate the inner world

A ( ) ==> A (A)   PERVASION

We generalize our ability to create reversible processes

((A)) ==> A   INVOLUTION

Completion

We accept these generalities as embodying our formal intentions.

We recognize that all rules are bidirectional.

DECOMPOSITION

The three axioms above decompose Spencer-Brown's two in this manner:

POSITION   ((p) p) =

( ) ==> ( ) A ==> (A) A

position = dominion + pervasion

TRANSPOSITION   ((p r) (q r)) = ((p)(q)) r

A ==> ((A)) ==> ((A)(A))

transposition = involution + replication

ALGEBRAIC VERIFICATION (PROOF)

Proofs with these axioms are usually elegant. We know specifically which steps introduce structure creatively, and which remove irrelevancies. Eg:

```
REPLICATION            A A   =   A

                       A A  ==>  A (   (A) )              enfold
                            ==>  A ( A (A) )              insert
                            ==>  A ( A ( ) )              extract
                            ==>  A (   ( ) )              absorb
                            ==>  A                        clarify
```

Transposition (aka distribution) becomes a theorem, on the same order as absorption, flex, and cancellation. Each are *heuristics* to untangle hard problems. Thus they can be applied with intelligence and discernment.

By suppressing sets of the axioms, different mathematics are formed:

No suppression	generates propositional calculus
( ) A =/= ( )	generates integer arithmetic
((A)) =/= A	generates finite set theory
A (A) =/= A ( )	generates imaginary logics

BOTTOM LINE

So the bottom line on Boolean axioms is

INVOLUTION	((A)) = A
DOMINION	( ) A = ( )
PERVASION	(A) A = ( ) A

TECHNICAL APPENDIX

The proof of distribution/transposition from the three axioms:

Lemma: REPLICATION from above

Lemma: ABSORPTION ((A)(A B)) = A

```
                    ( (A) (    A B) )
                ==> ( (A) ((A) A B) )                  insert (A)
                ==> ( (A) (( ) A B) )                  extract A
                ==> ( (A)           )                  absorb and
clarify
                ==>    A                               clarify
```

DISTRIBUTION/TRANSPOSITION ((A B)(A C)) = A ((B)(C))

```
          ( (A B) (   A                 C   ) )

      ==> ( (A B) ( (      (A)) (      (C)) ) )       enfold
      ==> ( (A B) ( ((A B) (A)) ((A B) (C)) ) )       insert
      ==> ( (A B) (         A   ((A B) (C)) ) )       absorption
      ==> ( (A B) (         A   ((  B) (C)) ) )       extract

      ==> ((   A                  B                ) (A ((B)(C))))   reformat
      ==> (( ((A)              ) ((B)             ) ) (A ((B)(C))))   enfold
      ==> (( ((A) (A ((B)(C)))) ((B) (A ((B)(C)))) ) (A ((B)(C))))   insert
      ==> ((   A                ((B) (A ((B)(C)))) ) (A ((B)(C))))   absorb
      ==> ((   A                ((B) (A (   (C)))) ) (A ((B)(C))))   extract
      ==> ((   A                ((B) (A      C  )) ) (A ((B)(C))))   clarify
      ==> ((   A                ((B) (       C  )) ) (A ((B)(C))))   extract

      ==> ((A ((B)(C)))                             (A ((B)(C))))   reformat
      ==> ((A ((B)(C)))                                         )   coalesce
      ==>   A ((B)(C))                                               clarify
```

ADDITION JANUARY 1987

The above proof is too clumsy, and the absorb lemma is not needed. Here's a prettier proof:

```
DISTRIBUTION            ((A B)(A C)) = A ((B)(C))

           (( A   B                ) ( A        C   ) )

      ==> ( ( A   B                ) ( A (     (C)) ) )     enfold
      ==> ( ( A   B                ) ( A ((A B)(C)) ) )     insert
      ==> ( ( A   B                ) ( A ((  B)(C)) ) )     extract

      ==> ( ( A ((B)             ) ) ( A ((  B)(C)) ) )     enfold
      ==> ( ( A ((B) (A ((B)(C)))) ) ( A ((  B)(C)) ) )     insert
      ==> ( ( A ((B) (A (   (C)))) ) ( A ((  B)(C)) ) )     extract
      ==> ( ( A ((B) (A      C  )) ) ( A ((  B)(C)) ) )     clarify
      ==> ( ( A ((B) (       C  )) ) ( A ((  B)(C)) ) )     extract
      ==> ( ( A ((B) (       C  )) )                  )     coalesce
      ==>     A ((B) (       C  ))                          clarify
```

ADDITION DECEMBER 1987

Here's another pretty proof, this one is shorter. It uses the DEEP ABSORPTION theorem.

```
DISTRIBUTION            A ((B)(C)) = ((A B)((A C))

            A                ((  B)(  C))

      ==>   A                ((A B)(A C))                   insert
      ==> ((A)             ) ((A B)(A C))                   enfold
      ==> ((A) ((A B)(A C))) ((A B)(A C))                   insert
      ==> ((A) (          )) ((A B)(A C))                   absorb
      ==>                    ((A B)(A C))                   occlude
```

ADDITION OCTOBER 1996

A variety of proofs of distribution, while experimenting with techniques for virtual insertion.

```
((a   b              )(a        c   ))          +involution
((a   b              )(a ((a b)(c)) ))          +pervasion
((a   b              )(a ((  b)(c)) ))          -pervasion
((a ((b)            ))(a ((  b)(c)) ))          +involution
((a ((b)(a ((b)(c)))))(a ((  b)(c)) ))          +pervasion
((a ((b)(  (   (c)))))(a ((  b)(c)) ))          -pervasion
((a ((b)(       c  )))(a ((  b)(c)) ))          -involution
((a ((b)(       c  )))               )          -pervasion
  a ((b)(       c  ))                           -involution
```

```
  a                            ((  b)(  c))
  a                            ((a b)(a c))     +pervasion
((a)                         ) ((a b)(a c))     +involution
((a) (a b)(a c)              ) ((a b)(a c))     +subsumption
((a) (a b)(a c) ((a b)(a c)) ) ((a b)(a c))     +pervasion
((a) (a b)(a c) (          ) ) ((a b)(a c))     -pervasion
                               ((a b)(a c))     -occlusion
```

```
  a                        ((  b)(  c))
  a                        ((a b)(a c))         +pervasion
((a)                     ) ((a b)(a c))         +involution
((a) ((    a b)(    a c))) ((a b)(a c))         +pervasion
((a) (((a) a b)((a) a c))) ((a b)(a c))         +pervasion
((a) ((( ) a b)(( ) a c))) ((a b)(a c))         -pervasion
((a) (                  )) ((a b)(a c))         -occlusion
                           ((a b)(a c))         -occlusion
```

Summarize as

```
  a                        ((  b)(  c))
  a                        ((a b)(a c))         +insertion
((a)                     ) ((a b)(a c))         +involution
((a) ((    a b)(    a c))) ((a b)(a c))         +reinsertion
                           ((a b)(a c))         -insertion
```

So the "a" which taken out and back in was never really there

```
  a                        ((  b)(  c))
((a)                     ) ((a b)(a c))         +involution
((a) ((      b)(      c))) ((a b)(a c))         +preinsertion
((a) ((    a b)(    a c))) ((a b)(a c))         +insertion
                           ((a b)(a c))         -insertion
```

The outside "a" is needed to build the target expression before it is blown away.

```
  a            ((b)(c))
((a)         ) ((a b)(a c))         +involution and insertion
((a) ((b)(c))) ((a b)(a c))         +preinsertion
((a) ((a b)(a c))) ((a b)(a c))     +strange insertion
                   ((a b)(a c))
```