Volume VIII

Tractatus Principio

—

On Difference

Volume VIII

Book of Application

by

C.H.F. Warnke

Table of Contents

Preface: The folding-back of the seven volumes
Chapter I: From difference to Delta
Chapter II: From Delta to number
Chapter III: From number to fractal number
Chapter IV: From number to operation
Chapter V: From operation to geometry
Chapter VI: From geometry to observer
Chapter VII: From observer to trace
Chapter VIII: From trace to reading
Chapter IX: The isomorphy of reading
Chapter X: Rules of application
Appendix: The sign evolution and note on use
Index: Quick register
Index: Basic formulas at a glance
Index: Symbol overview of the Tractatus Principio
Part II: From the sign to the executable sign (Python as representation of difference)
Preface: The second writing
Chapter X: Code as inscription
Chapter XI: The three layers of code
Chapter XII: The stroke in code
Chapter XIII: Delta as transition
Chapter XIV: Number and mark
Chapter XV: Operators as modes of binding
Chapter XVI: Variable and opening
Chapter XVII: Gate and visibility
Chapter XVIII: Path, loop, and trace
Chapter XIX: Observer and context in code
Chapter XX: Projection, binding, and equality
Chapter XXI: Normal form and readout
Chapter XXII: Error as relational error
Chapter XXIII: A complete double reading

Chapter XXIV: Python as second notation

Chapter XXV: Basic rules of Delta-constraint coding

Conclusion: The executable sign

Preface

The folding-back of the seven volumes

Volume VIII introduces no new theory.

It folds the seven preceding volumes back into an operative sign sequence.

$$|\rightarrow \Delta \rightarrow 0\text{–}9 \rightarrow \mathit{Operation} \rightarrow \mathit{Geometry} \rightarrow \mathit{Observer} \rightarrow \mathit{Trace} \rightarrow \mathit{Reading}$$

The signs are not grounded anew.

They are set in their application.

The formula is not read as an object.

It is used as enactment of difference.

The evolution of signs and the isomorphy of reading

Chapter I

From difference to Delta

$$|$$

The difference of first order.

$$A|B$$

The difference of second order.

$$(A|B)|(C|D)$$

The difference of third order.

$$|(|)$$

The self-referentiality of difference.

$$\Delta = |(|)$$

Delta as smallest effective self-referentiality.

Delta as beat.

Delta as transition.

Delta as marking.

Delta as self-rotation.

Delta as drift.

No classes of Delta.

No separate kinds of difference.

The multiplicity appears in the third order of the same relation.

Chapter II

From Delta to number

$$\Delta \to 0|1|2|3|4|5|6|7|8|9$$

The number as stabilized reading of Delta.

$$n := read(\Sigma_n)$$

There are only the numbers:

$$\boxed{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}$$

The numbers as visible basic marks of a complete shell.

$$\mathbf{0}$$

closed possibility.

$$\mathbf{1}$$

first marking.

$$\mathbf{2}$$

mirror relation.

$$\mathbf{3}$$

orientation.

$$4$$

context stabilization.

$$5]self - similarity.\,[6$$

division resonance.

$$7$$

drift.

$$8$$

double fourfold folding.

$$9$$

break edge.

$$9 := \kappa$$

Chapter III

From number to fractal number

$$9 \to 1|0][10 := 1|0$$

Ten is not an eleventh number.

Ten is the first visible relation of a new reading shell.

Multi-digit numbers as fractal inscriptions of the same ten marks.

$$125 := 1|2|5][2026 := 2|0|2|6$$

The place is not a container.

The place is relational depth.

$$\Delta_2|\Delta_1|\Delta_0$$

Horizontal effect within a shell.

Vertical effect between shells.

Feedback effect of lower shells on higher shells.

Carry as break-edge feedback effect.

$$199 \to 200$$

Digit sum as basic-shell reading.

Decimal point as scale flip.

$$|.$$

Negative number as reading direction.

$$-\Delta| + \Delta$$

Chapter IV

From number to operation

The number marks do not act on one another as objects.

They enter into modes of difference.

Expansion:

$$|_{+}$$

Contraction:

$$|_{-}$$

Resonance:

$$|_{\times}$$

Disentangling:

$$|_{\div}$$

Exponentiation:

$$|$$

Rooting:

$$|_{\sqrt{\ }}$$

Equilibrium:

$$|_{=}]Non-coincidence:[|_{\neq}$$

Differential:

$$|_{\Delta}$$

Integral:

$$|\Sigma|\Delta$$

The operator produces no value.

It orients the relation.

$$a|_{+}b][a|_{-}b][a|_{\times}b][a|_{\div}b][a|_{=}b$$

Calculation as ordered inscription.

$$I = (m_0, m_1, \ldots, m_n)$$

Chapter V

From operation to geometry

Every operation possesses a geometrical reading.

$$0|1$$

Radius.

$$0|2$$

Diameter and axis.

$$0|3$$

Triangle and orientation.

$$0|4$$

Square and stable fourfold order.

$$0|5$$

Pentagon and self-similarity.

$$0|6$$

Hexagon and division resonance.

$$0|7$$

Heptagon and drift.

$$0|8$$

Octagon and double fourfold folding.

$$0|9$$

Nonagon and break edge.

The number is geometrical relation.

Geometry is number reading.

$$Number \cong Geometry][\pi$$

as relation of closed and linear reading.

$$\varphi$$

as self-similarity relation.

$$e$$

as stable self-reference of the rate of change.

$$39{,}19$$

as self-reference constant of the number relation.

$$i := R_{90^\circ}$$

as rotation address.

$$\Delta = 2{,}25^\circ$$

as smallest self-rotation.

Chapter VI

From geometry to observer

Geometry is not read from outside.

The reading possesses an orientation.

Frozen frame:

$$|_|$$

Marked frame:

$$|\Delta$$

The Delta states:

$$0\Delta$$

Frame.

$$1\Delta$$

Event.

$$2\Delta$$

Axis.

$$3\Delta$$

Observer.

$$4\Delta$$

full context.

$$4\Delta = \Delta^{\Delta}$$

The observer as oriented incompleteness of the context.

$$3\Delta = 4\Delta - \Delta$$

Hole direction:

$$h(o)$$

Orientation space:

$$\mathcal{O}$$

Transition space:

$$\mathcal{T}$$

Time as sequence of orientation changes.

$$o_0 \to o_1 \to o_2 \to \dots$$

Chapter VII

From observer to trace

The inscription is read as path.

$$I \to \gamma_I$$

The path produces no number.

It carries a trace.

$$\gamma_I \to \Sigma(\gamma_I)][\Sigma(\gamma) := \sum_{e \in \gamma} w(e)$$

Visibility:

$$L_n$$

Gate:

$$\chi_{\mathrm{vis}}][\chi_{\mathrm{gate}}$$

Step contribution:

$$H_n := \chi_{\mathrm{vis}}(e_n)\chi_{\mathrm{gate}}(e_n)w(e_n)$$

Trace accumulation:

$$\Sigma_{n+1} = \Sigma_n + H_n$$

The trace as memory of the relation.

Holonomy as difference of the return.

$$\gamma(0) = \gamma(1)$$

and nevertheless:

$$\Sigma(\gamma) \neq 0$$

Return to the starting place is not return to the starting position.

Chapter VIII

From trace to reading

The trace is projected.

$$\Pi_i(\Sigma)$$

The projections are bound.

$$\bowtie$$

The drift is read.

$$\Delta\varphi$$

The relation locks in.

$$Lock_{\frac{3}{4}}$$

The trace forms a normal form.

$$NF(\Sigma)$$

The normal form becomes publicly readable.

$$read(NF(\Sigma))$$

The complete enactment of reading:

$$I \to \gamma_I \to \Sigma \to \Pi \to \bowtie \to Lock_{\frac{3}{4}} \to NF \to read$$

The result is not a produced object.

$$Ergebnis := read(NF(\Sigma))$$

Equality is not identity of things.

$$|_{=}$$

denotes the locking-in of the same trace under a different reading.

$$A|_{=}B][\Sigma_A \equiv_\Sigma \Sigma_B$$

Chapter IX

The isomorphy of reading

The signs do not designate separate worlds.

They are different readings of the same order of difference.

$$|\cong \Delta \cong Number \cong Geometry \cong Observer \cong Trace \cong Reading$$

Not as equality of objects.

Not as interchangeability of the signs.

But as preservation of the same relational structure.

$$|\to \Delta \to 0\text{–}9 \to I \to \gamma \to \Sigma \to NF \to read$$

and backward:

$$read \to NF \to \Sigma \to \gamma \to I \to 0\text{–}9 \to \Delta \to |$$

The read-back preserves the trace class.

$$recover(read(\Sigma)) \equiv_{\Sigma} \Sigma$$

Validity:

$$valid(L) \iff recover(L) \equiv_{\Sigma} \Sigma$$

The forms change.

The relation remains readable back.

The signs develop.

The reading remains isomorphic.

Chapter X

Rules of Application

1. *No sign denotes a thing.* §1. *Every sign is a reading position of the difference.* §2. *A form no absence of itself.* §3. *There are only the numbers 0 to 9.* §4. *Mark = difference to frontal relation.* §5. *0 is the frontier sign.* §7. *10 = 1|0.* §8. *Operators align relations.* §9. *Calculation is inscription.* §10. *Inscription becomes path.* §11. *Path becomes trace.* §12. *Trace is read.* §13. *Equality is looking-as.* §14. *The result is readout.* §15. *Every readout must remain readable back to the trace.* §16. *Different readings of the same trace are isomorphic.*

Appendix

The Sign Evolution

|][↓][A|B][↓][|(())|][↓][Δ][↓][0|1|2|3|4|5|6|7|8|9][↓][1|0][↓][I][↓][γ][↓][Σ][↓][NF][↓][$read$][$\boxed{|\cong \Delta \cong Number \cong Geometry \cong Observer \cong Trace \cong Reading}$]

Usage Note

How this book is to be used

This book is a formula collection of the Tractatus Principio.

It introduces no new theory.

It does not ground the signs anew.

It gathers the basic forms, operators, transitions, and reading rules of the seven preceding volumes in a common operative order.

The entries are not to be read as definitions separated from one another.

Every sign receives its meaning from its position in the complete enactment:

$$|\to \Delta \to 0\text{–}9 \to Operation \to Geometry \to Observer \to Trace \to Reading$$

The formulas denote no objects.

They set reading positions of the difference.

A formula is used by reading its signs as relational positions, transferring them into a path, integrating them as a trace, and bringing them into a read-back capable normal form.

The basic enactment is:

$$\boxed{I \to \gamma_I \to \Sigma(\gamma_I) \to \Pi_i(\Sigma) \to \bowtie \to Lock_{\frac{3}{4}} \to NF(\Sigma) \to read}$$

Here the following holds:

$$\boxed{\textbf{It is not things that are processed.}} \;] [\; \boxed{\textbf{Differences are set, enacted, bound, and read.}}$$

Quick Index

By Basic Concepts

Sought

```
        Sign
        Section
```

Difference

$$|$$

Poles of a relation

$$A|B$$

$$Self-referentiality[|(|)]$$

```
        Delta
        \Delta
        Basic Formulas I
        Possibility
        \odot
        Basic Formulas II
        Null
```

```
        0
        Basic Formulas II
        Impossibility
        \boxdot
        Basic Formulas II
        Drift
        \Delta\varphi
        Basic Formulas II
        Number
        n:=read(\Sigma_n)
        Basic Formulas III
        Fracture edge
        9:=\kappa
        Basic Formulas III
```

new reading shell

$$10 := 1|0$$

```
        multi-digit number
        a
        b
```

negative number

$$-\Delta|+\Delta$$

Addition

$$|_{+}$$

Subtraction

$$|_{-}$$

Multiplication

$$|_{\times}$$

Division

$$|_{\div}$$

Equality

$$|_{=}$$

$$Non-coincidence[|_{\neq}]$$

Differential

$$|_{\Delta}$$

Integral

$$|\Sigma|_{\Delta}$$

Frame

$$|_{_}|$$

```
Observer
3\Delta
Observer
Context
4\Delta
Observer
Hole direction
h(o)
Observer
Inscription
I
Reading enactment
Path
\gamma_I
Reading enactment
Edge
e_n
Reading enactment
Gate
\chi_{\mathrm{gate}}
Trace
Visibility
\chi_{\mathrm{vis}}
Trace
Step contribution
H_n
Trace
Trace
\Sigma(\gamma)
Trace
Holonomy
\Omega(n)
Holonomy
Projection
\Pi_i
```

```
        Projection
        Binding
        \bowtie
        Projection
        Shaking space
        \mathcal L_{\frac34}
        ¾-corner
        Locking-in
        Lock_{\frac34}
        ¾-corner
        Normal form
        NF(\Sigma)
        Reading
        Readout
        read(\Sigma)
        Reading
        Reading back
        recover
        Reading
        Validity
        valid(L)
        Reading
        Trace-class equality
        \equiv_\Sigma
        Reading
```

By Application

Application

```
        Basic form
```

Set relation

$$x|y$$

Extend relation

$$x|_+y$$

Contract relation

$$x|_-y$$

Couple rhythms

$$x|_\times y$$

Unfold total figure

$$x|\div y$$

Iterate repetition

$$x|^{y}$$

Read iteration back

$$|\sqrt{\ }$$

Check locking-in

$$x|=y$$

Mark non-coincidence

$$x|\neq y$$

Read local change

$$|\Delta$$

Accumulate change

$$|\Sigma|\Delta$$

Change scale

$$|.$$

```
        Reverse reading direction
        -\Delta
        Form inscription
        I=(m_0,\dots,m_n)
        Clock inscription
        I\rightarrow\gamma_I
        Integrate trace
        \Sigma(\gamma)
        Bind projections
        \Pi_i\bowtie\Pi_j
        Read minimal drift position
        L^*
        Form normal form
        NF(\Sigma)
        Display result
        read(NF(\Sigma))
```

```
        Check result
        recover(read(\Sigma))\equiv_\Sigma\Sigma
```

Basic Formulas at a Glance

I. Difference and Self-Referentiality

Difference of the first order

$$\boxed{|}$$

Pure difference.

No executed poles yet.

The difference separates no already existing objects.

It opens the possibility of relational position.

Difference of the second order

$$\boxed{A|B}$$

The difference appears in two relational poles.

A and B do not exist before their relation.

They are positions of the same difference.

Difference of the third order

$$\boxed{(A|B)|(C|D)}$$

or:

$$\boxed{|(|)}$$

The difference differentiates its own differences.

Relational depth arises.

Delta

$$\boxed{\Delta = |(|)}$$

Delta is the smallest effective self-referentiality of the difference.

Delta appears as:

$$\Delta \cong \textit{Beat} \cong \textit{Transition} \cong \textit{Marking} \cong \textit{Rotation} \cong \textit{Drift}$$

These expressions denote no different kinds of Delta.

They are different operative readings of the same smallest referentiality.

II. Possibility, Mirroring, and Impossibility

Possible Geometry



Closed possibility.

Read-back capable unfolding of a relation.

Null



Null is not nothingness.

Null is the closed cutting, mirroring, and equilibrium position of the difference.



Not as interchangeability of the signs, but as isomorphic reading of the same relational basic structure.

Impossible Geometry

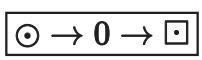


Open, drifting, or not yet locked-in relation.

The impossible denotes no second object.

It denotes the not-yet-closed position of a relation.

Complete Process



Unfolding, mirroring, and testing.

Locking-in does not lead back to the old possibility.

$$\boxed{\boxdot \to |_{=} \to \odot'}$$

The new closed possibility carries the trace of the enactment it has undergone.

Drift

$$\boxed{\Delta\varphi \neq 0}$$

Difference movement not yet locked in.

Drift is not a deviation from an externally given norm.

It is the still open difference between possible relational positions.

III. Number and Reading Shell

Number

$$\boxed{n := read(\Sigma_n)}$$

A number is the visible reading of a stabilized trace.

The number is not an original object.

It is a public readout.

Complete mark space

$$\boxed{M := 0,1,2,3,4,5,6,7,8,9}$$

There are only the ten visible basic marks:

$$\boxed{0,1,2,3,4,5,6,7,8,9}$$

All multi-digit numbers are relational inscriptions of the same marks.

Number constitution

$$0$$

Closed possibility.

$$1$$

First marking.

2

Mirror relation.

3

Orientation.

4

Context stabilization.

5

Self-similarity opening.

6

Division resonance.

7

Drifting closure.

8

Double fourfold folding.

9

Break edge and cycle closure.

Break edge

$$\boxed{9 := \kappa}$$

The nine closes the visible basic shell.

It is at the same time the edge at which the reading passes into a new shell.

New reading shell

$$\boxed{9 \to 1|0}][\boxed{10 := 1|0}$$

Ten is not an eleventh basic mark.

Ten is the first visible relation of a new reading shell.

Multi-digit number

$$\boxed{125 := 1|2|5}\,]\,[\boxed{2026 := 2|0|2|6}$$

A multi-digit number is an ordered fractal inscription.

The place is not a container.

It is relational depth.

Levels of a number

$$\boxed{\Delta_2|\Delta_1|\Delta_0}\,]\,[\Delta_0$$

immediate beat and ones shell.

$$\Delta_1$$

first mirroring and tens shell.

$$\Delta_2$$

meta-relation and hundreds shell.

The levels do not designate different kinds of numbers.

They designate different relational positions of the same marks.

Carry

$$\boxed{199 \to 200}$$

Carry is break-edge retroaction.

A closed lower shell changes the position of the higher shell.

Decimal point

$$\boxed{|.}$$

The decimal point is a scale flip.

It does not separate two number worlds.

It changes the direction in which relational depth is read.

Negative number

$$\boxed{-\Delta|+\Delta}$$

Negative and positive designate reading directions.

Negative numbers do not form a second number world.

$$\boxed{-1|0|1}$$

is the complete axis relation of negative reading direction, zero, and positive reading direction.

Operators

Basic rule

$$\boxed{\textbf{The operator does not generate a value.}}\ \boxed{\textbf{The operator aligns the relation.}}$$

The visible kind of calculation designates the mode in which number marks are related to one another.

Expansion

$$\boxed{x|_{+}y}$$

x expands by y.

A relation is opened by a further relation.

Public reading:

$$x + y$$

Contraction

$$\boxed{x|_{-}y}$$

x contracts by y.

A relation is led back to a narrower position.

Public reading:

$$x - y$$

Resonance

$$\boxed{x|_{\times}y}$$

x resonates with y.

Relations couple their levels, repetitions, or rhythms.

Multiplication is not merely repeated addition.

Public reading:

$$x \times y$$

Disentanglement

$$\boxed{x|_{\div}y}$$

x is disentangled by y.

An overall figure is read into recurring or supporting relations.

Public reading:

$$x \div y$$

Exponentiation

$$\boxed{x|^{y}}$$

A relational effect itself becomes the carrier of its repetition.

Exponentiation is iterated iteration.

Rooting

$$\boxed{|_{\sqrt{\ }}}$$

Back-reading of an iterated relation onto its supporting repetition figure.

Equilibrium

$$\boxed{x|_{=}y}$$

The relation between x and y reaches a stable position under a certain reading.

Equality is not an absolute identity of things.

It is the locking-in of the same trace under different readings.

$$Non-coincidence \boxed{\boxed{x|_{\neq}y}}$$

The two readings do not lock in under the given position.

Non-coincidence does not generate a positive object.

It keeps the open or impossible relation space visible.

Impossible distinction

$$\boxed{x|_{\triangledown}y}$$

The relation remains unstable, open, or drifting under the current reading.

\triangledown does not designate a second kind of Delta.

It designates Delta in a position not yet locked in.

Differential

$$\boxed{|_{\Delta}}$$

Smallest self-movement.

Local change in the immediate beat.

Integral

$$\boxed{|\Sigma|_{\Delta}}$$

Accumulated self-movement.

Integrated difference along a path.

Geometric basic readings

Basic rule

$$\boxed{\text{Number} \cong \text{Geometry}}$$

Not as equality of two objects.

Number and geometry are different readings of the same relational order.

0|1

Radius relation.

0|2

Diameter and axis.

0|3

Triangle and orientation.

0|4

Square and stable fourfold order.

0|5

Pentagon and self-similarity relation.

0|6

Hexagon and closed division resonance.

0|7

Heptagon and drifting division relation.

0|8

Octagon and double fourfold folding.

0|9

Nonagon and break edge of the current shell.

Rotation address

$$\boxed{i := R_{90^\circ}}\boxed{-i := R_{-90^\circ}}$$

i is not an additional kind of number.

i is the address of a quarter rotation of the same relational figure.

$$i^2 := R_{180^\circ}][i^3 := R_{270^\circ}][i^4 := id$$

Frame, observer, and context

Frozen frame

$$\boxed{F := |_|}$$

Closed local possibility.

Marked frame

$$\boxed{|_\Delta}$$

Frame with set Delta marking.

Direction set

$$\boxed{D := N, E, S, W}$$

Delta configuration

$$\boxed{o : D \to 0, 1}$$

o designates the set and open Delta positions around a frame.

Number of set positions

$$\boxed{k(o) := \sum_{d \in D} o(d)}$$

Delta states

$$0\Delta$$

Pure frame.

No set Delta position.

$$1\Delta$$

Event.

One set position.

$$2\Delta$$

Axis.

Two set positions.

$$3\Delta$$

Observer orientation.

Three set positions and one open direction.

$$4\Delta$$

Full context space.

Four set positions.

Context

$$\boxed{4\Delta = \Delta^{\Delta}}$$

The full context is the saturated Delta configuration.

Observer

$$\boxed{3\Delta = 4\Delta - \Delta}$$

The observer is the oriented incompleteness of the context.

It is not a position outside the relation.

It is a relational position with an open direction.

Hole direction

$$\boxed{h(o)}$$

Open direction of a three-Delta state.

Epistemic outside of the observer.

Orientation space

$$\boxed{\mathcal{O} = o_N, o_E, o_S, o_W}$$

Transition space

$$\boxed{|\mathcal{T}| = 12}$$

Twelve directed transitions between four observer orientations.

Time

$$\boxed{o_0 \to o_1 \to o_2 \to \dots}$$

Time is the readable sequence of orientation changes.

Inscription, path, and beat

Inscription

$$\boxed{I = (m_0, m_1, \dots, m_{N-1})}$$

Ordered sequence of visible marks and operator edges.

A calculation is an inscription.

Path of the inscription

$$\boxed{\gamma_I = (e_0, e_1, \dots, e_{N-1})}$$

Clocked sequence of the operative transitions of an inscription.

Edge

$$\boxed{e_n : x_n \to x_{n+1}}$$

An operative step within a path.

Clocked reading operation

$$\boxed{T_n := \mathit{Step} \circ O_n}$$

$$Non-commutativity \boxed{Step \circ O \neq O \circ Step}$$

The order of step and reading operation carries meaning.

A changed order is a changed inscription of the enactment.

Trace and visibility

Difference reading of an edge

$$\boxed{w(e)}$$

Relational contribution of an edge to the trace.

Visibility

$$\boxed{\chi_{\text{vis}}(e) \in 0,1}$$

Determines whether an edge is visible under the present reading.

Gate

$$\boxed{\chi_{\text{gate}}(e) \in 0,1}$$

Determines whether a visible edge is admitted to the present enactment.

Step contribution

$$\boxed{H_n := \chi_{\text{vis}}(e_n) \cdot \chi_{\text{gate}}(e_n) \cdot w(e_n)}$$

An edge contributes to the trace only when it is visible and admitted.

Trace

$$\boxed{\Sigma(\gamma) := \sum_{e \in \gamma} w(e)}$$

Integrated difference of a path.

The trace is the memory of the enacted relation.

Trace accumulation

$$\boxed{\Sigma_{n+1} = \Sigma_n + H_n}$$

Every admitted step changes the readable history of the path.

Readable edges

$$\boxed{L_n}$$

Set of the edges visible and admitted in beat n.

Holonomy

Accumulated holonomy

$$\boxed{\Omega(n) := \sum_{t=0}^{n-1} H(t)}$$

Phase position

$$\boxed{\phi(n) := \Omega(n) \bmod 2\pi}$$

Non-vanishing return difference

$$\boxed{\gamma(0) = \gamma(1)}$$

and at the same time:

$$\boxed{\Sigma(\gamma) \neq 0}$$

Return to the starting place is not necessarily return to the starting position.

The closed path can carry a non-vanishing trace.

Projection and binding

Projection

$$\boxed{\Pi_i : \Sigma \to A_i}$$

A reading channel of a trace.

Projected readout

$$\boxed{a_i := \Pi_i(\Sigma)}$$

Difference of a reading channel

$$\boxed{\Delta a_i(n) := a_i(n) | a_i(n+1)}$$

Binding gate

$$\boxed{\Delta a_i \bowtie \Delta a_j}$$

The binding gate checks the compatibility of different trace readings.

Binding does not mean fusion.

It denotes the ability of different projections to lock in together.

The ¾-corner

¾-corner

$$\boxed{\Theta_{\frac{3}{4}} := (e_1, e_2, e_3; D)}$$

Three set hinge edges and one open drift edge.

Jostling space

$$\boxed{\mathcal{L}_{\frac{3}{4}}}$$

Finite space of possible locking-in positions.

Minimal drift position

$$\boxed{L^* := \underset{L \in \mathcal{L}_{\frac{3}{4}}}{\arg\min} drift(L)}$$

Equality snap

$$\boxed{Lock_{\frac{3}{4}}}$$

Finite locking-in of a relation in its minimal drift position.

The ¾-corner does not generate the result.

It determines the position in which the bound projections become readable as a common normal form.

Normal form and readout

Normal form

$$\boxed{NF(\Sigma)}$$

Self-resonance reading of a trace.

Stably positioned readable-back state.

Public reading

$$\boxed{read(\Sigma)}$$

Visible readout of a trace.

Result

$$\boxed{Ergebnis := read(NF(\Sigma))}$$

The result is not a newly generated object.

It is the public reading of a locked-in trace.

Trace-class equality

$$\boxed{G \equiv_{\Sigma} I}$$

G and I carry the same back-addressable trace class.

Resonance comparison

$$\boxed{\sim}$$

Resonance similarity without trace-class equality.

Reading transition

$$\boxed{\rightsquigarrow}$$

A form falls under a particular reading onto another visible form.

$$Back - reading \llbracket recover \rrbracket$$

Reconstruction of a trace from its readout.

Validity

$$\boxed{valid(L) \iff recover(L) \equiv_{\Sigma} \Sigma}$$

A visible label is valid if it remains back-readable onto its trace class.

Application templates

Basic schema of an application

$$\boxed{\textbf{form inscription} \to \textbf{read path} \to \textbf{integrate trace} \to \textbf{bind projections} \to \textbf{minimize drift} \to \textbf{read normal form}}$$

Formal:

$$\boxed{I \to \gamma_I \to \Sigma \to \Pi \to \bowtie \to Lock_{\frac{3}{4}} \to NF \to read}$$

Addition

Public inscription

$$a + b$$

Operative inscription

$$\boxed{a|_{+}b}$$

Reading schema

$$a| + b \to I \to \gamma_I \to \Sigma \to Lock\frac{3}{4} \to NF \to read$$

Mnemonic sentence

Addition is expansion of a relation.

Subtraction

Public inscription

$$a - b$$

Operative inscription

$$\boxed{a|_b}$$

Reading schema

$$a|_b \to \textbf{contraction} \to \Sigma \to NF \to read$$

Mnemonic sentence

Subtraction is contraction or reversal of a reading direction.

Multiplication

Public inscription

$$a \times b$$

Operative inscription

$$\boxed{a|_{\times} b}$$

Reading schema

$$a| \times b \to \textbf{rhythm coupling} \to \Sigma \to Lock\frac{3}{4} \to read$$

Mnemonic sentence

Multiplication is resonance of different relational rhythms.

Division

Public inscription

$$a \div b$$

Operative inscription

$$\boxed{a|_{\div} b}$$

Reading schema

$$a|\div b \to \textbf{disentanglement} \to \Pi_i(\Sigma) \to NF \to read$$

Mnemonic sentence

Division reads a total figure back to its supporting repetition relations.

Equation

Public inscription

$$A = B$$

Operative inscription

$$\boxed{A|_{=}B}$$

Condition

$$\boxed{\Sigma_A \equiv_{\Sigma} \Sigma_B}$$

Enactment

$$read_L \to \Sigma_L][read_R \to \Sigma_R][\Sigma_L \bowtie \Sigma_R][Lock_{\frac{3}{4}}][NF$$

Mnemonic sentence

Equality is the locking-in of different readings of the same trace class.

Multi-digit number

Inscription

$$\boxed{a|b|c}$$

Layer reading

$$\boxed{\Delta_2|\Delta_1|\Delta_0}$$

Mnemonic sentence

Multi-digitness is not a collection of number objects.

It is relational depth of the same ten basic marks.

Negative number

Inscription

$$\boxed{-\Delta|+\Delta}$$

Mnemonic sentence

The sign denotes the direction of the reading.

It generates no second kind of number.

Geometric reading

Inscription

$$\boxed{0|n}$$

Assignment

$$0|1 \to radius][0|2 \to axis][0|3 \to orientation][0|4 \to fourfoldorder][0|5 \to self-similarity][0|6 \to divisionresonance][0|7 \to drift][0|8 \to doublefourfoldfolding][0|9 \to breakedge$$

Observer change

Starting position

$$o_n \in \mathcal{O}$$

Transition

$$\boxed{o_n \to o_{n+1}}$$

Sequence

$$o_0 \to o_1 \to o_2 \to \dots$$

Mnemonic sentence

An observer change changes the orientation of the reading.

It generates no observer outside the relation.

Holonomic reading

Closed path

$$\gamma(0) = \gamma(1)$$

Trace check

$$\Sigma(\gamma) \neq 0$$

Mnemonic sentence

A closed path can carry an open return difference.

Back-readability check

Public form

$$L := read(\Sigma)]Back - reading[recover(L)$$

Check

$$\boxed{recover(L) \equiv_{\Sigma} \Sigma}$$

Result

$$valid(L)$$

Mnemonic sentence

A result is valid if its visible form can be traced back to the enacted difference.

Entry form for further formulas

Every new entry of this handbook follows the same order.

Sign

$$\boxed{\textbf{Formula}}$$

Name

Brief designation of the formula or operation.

Reading

What is read relationally in this formula?

Function

What change, binding, orientation, or closure is enacted?

Public form

How does the formula appear in ordinary mathematical or linguistic notation?

$$Back - reference$$

From which volume and which conceptual context does the formula arise?

Hierarchy of Formulas

A. Basic formulas

Basic formulas belong to the architecture of the Tractatus Principio itself.

$$|][A|B][|(|)][\Delta = |(|)][0 \cong |][n := read(\Sigma_n)][\Sigma(\gamma)][recover(read(\Sigma)) \equiv_\Sigma \Sigma$$

They are not bound to a single model.

B. Transition formulas

Transition formulas describe the enactment between different forms of reading.

$$I \to \gamma_I][\gamma_I \to \Sigma][\Sigma \to \Pi_i][\Pi_i \to \bowtie][\bowtie \to Lock_{\frac{3}{4}}][Lock_{\frac{3}{4}} \to NF][NF \to read$$

C. Model formulas

Model formulas concretize the basic architecture in a specific operative representation.

These include in particular:

$$D = N, E, S, W][o : D \to 0, 1][k(o) = \sum_{d \in D} o(d)][\mathcal{O} = o_N, o_E, o_S, o_W][|\mathcal{T}| = 12][\Theta_{\frac{3}{4}} = (e_1, e_2, e_3; D)][L^* := \arg\min_{L \in \mathcal{L}_{\frac{3}{4}}} drift(L)$$

Model formulas are reading tools that have emerged from the basic architecture.

They do not replace the basic formulas.

Back-references into Volumes I to VII

Volume I – Book of Mathematics

Look up under:

- nature of number,
- rhythm of number,
- operation,
- negative number,
- infinity,
- differential and integral,
- matrix of self-referentiality,
- unfolding of \Delta_x.

Basic references:

$$n := read(\Sigma_n)][-\Delta| + \Delta][|_\Delta][|\Sigma|_\Delta$$

Volume II – Book of Physics

Look up under:

- relation instead of substance,
- energy,

- mass,
- gravitation,
- time,
- Frozen Frame,
- spacetime,
- self-reference constant.

Basic references:

$$|_|][\Delta\varphi][39{,}19$$

Volume III – Book of Self-Knowledge

Look up under:

- observer,
- emergence of the I,
- memory,
- will,
- time,
- language,
- self-organization,
- self as relational order.

Basic references:

$$3\Delta = 4\Delta - \Delta][h(o)][\Sigma$$

Volume IV – Book of Self-Referentiality

Look up under:

- orders of difference,
- fractality,
- oscillation,
- rotation,
- indeterminacy,
- fractal operators.

Basic references:

$$|(|)][\Delta = |(|)][\ominus][\circledast][\mathcal{R}$$

Volume V – Book of the Smallest Relationality of Delta

Look up under:

- frame and Delta,
- zero- to four-Delta states,
- observer and context,
- orientations,
- transitions,
- smallest beat,
- geometry of number.

Basic references:

$$F := |_|][o : D \to 0, 1][3\Delta = 4\Delta - \Delta][4\Delta = \Delta^{\Delta}$$

Volume VI – Book of Holonomy

Look up under:

- lattice,
- edges,
- gate,
- rotation,
- loop,
- trace,
- holonomy,
- return difference.

Basic references:

$$\gamma_I][H_n][\Sigma(\gamma)][\Omega(n)][\gamma(0) = \gamma(1), \qquad \Sigma(\gamma) \neq 0$$

Volume VII – Book of Reading

Look up under:

- number fractal,
- inscription,
- operator-edges,
- projection,
- binding,
- ¾-corner,
- normal form,
- readout,
- read-backability.

Basic references:

$$9 := \kappa][10 := 1|0][I \to \gamma_I \to \Sigma \to \Pi \to \bowtie \to Lock_{\frac{3}{4}} \to NF \to read$$

Rules of Application

1

No sign designates a thing.

2

Every sign is a reading-position of difference.

3

The poles of a relation do not exist before their distinction.

4

Delta forms no classes of itself.

5

Multiplicity appears in the orders of the same relation.

6

There are only the number marks 0 to 9.

7

Multi-digitness is fractal relation.

8

Nine is the fracture-edge of the visible basic shell.

9

Ten is 1|0, not an eleventh basic mark.

10

An operator produces no value.

It orients a relation.

11

A calculation is an ordered inscription.

12

An inscription is read as a path.

13

A path carries a trace.

14

The trace is the memory of the enactment.

15

Visibility and gate determine which edges contribute to the trace.

16

A closed movement can carry a non-vanishing return difference.

17

A trace can appear in different projections.

18

Binding checks whether different projections can lock in together.

19

Equality is not identity of objects.

Equality is locking-in.

20

The result is the public readout of a normal form.

21

Every valid readout must remain read-backable to its trace class.

$$\boxed{22}$$

Different readings of the same trace are isomorphic.

Closing Sheet

The Evolution of Signs

$$|][\downarrow][A|B][\downarrow][|(|)][\downarrow][\Delta][\downarrow][0|1|2|3|4|5|6|7|8|9][\downarrow][1|0][\downarrow][I][\downarrow][\gamma_I][\downarrow][\Sigma][\downarrow][NF][\downarrow][read$$

The Isomorphy of Reading

$$\boxed{|\cong 0 \cong \Delta \cong number \cong geometry \cong observer \cong trace \cong reading}$$

Not as equality of objects.

Not as interchangeability of signs.

But as preservation of the same relational structure under different readings.

The Complete Outward Path

$$\boxed{|\to \Delta \to 0\text{–}9 \to I \to \gamma_I \to \Sigma \to NF \to read}$$

The Complete Return Path

$$\boxed{read \to NF \to \Sigma \to \gamma_I \to I \to 0\text{–}9 \to \Delta \to |}$$

$$Read-backability[\boxed{recover(read(\Sigma)) \equiv_\Sigma \Sigma}]$$

The signs change.

The trace remains read-backable.

The readings differ.

The relational order remains isomorphic.

Principle of Application

$$\boxed{\text{Do not calculate in order to produce an object.}}][\boxed{\text{Distinguish, sequence into beats, bind, and read.}}$$

Principle of the Handbook

As little repetition as possible.][**As much orientation as necessary.**][**As much read-backability as required.**

Symbol Overview of the Tractatus Principio

A reference of the introduced notations

I. Basic Symbols

| – The difference

Fundamental operation from which everything arises.

$$|$$

The difference in open positing.

It does not subsequently separate already existing things.

Through it the poles of a relation arise.

$$|\rightarrow A|B$$

---

0 – Zero / Potential

The difference itself as closed cut, mirror, and equilibrium position between possible and impossible geometry.

$$\odot|0|\boxdot$$

Zero is not nothing.

Zero is the closed possibility of difference.

$$0 \cong |$$

Not as interchangeability of signs, but as isomorphic reading of the same difference.

$$\Delta – Delta$$

Smallest self-referentiality and smallest self-rotation of difference.

$$\Delta = |(|)][\Delta = 2{,}25°$$

Delta appears as:

- beat,
- transition,
- marking,
- self-rotation,
- drift,
- decision,
- smallest efficacy of difference.

These appearances are not different Delta kinds.

They are relational readings of the same \Delta.

$$\Delta\varphi – Delta – Phi$$

Free difference energy.

Drift between relational positions.

$$\Delta\varphi \neq 0$$

designates a difference movement not yet locked in.

II. The Three Orders of Difference

First order

$$|$$

Pure difference.

As yet no executed poles.

Second order

$$A|B$$

Polarity.

Dichotomy.

Two relational positions of the same difference.

$$A \neq B$$

The poles do not exist before their relation.

Third order

$$(A|B)|(C|D)$$

or:

$$|(|)$$

The difference distinguishes its own differences.

The third order does not produce mere multiplication of elements.

It produces relational depth.

$$A|B|C|\ldots$$

is a possible unfolded notation, insofar as differences act on differences.

III. Three Phases of Self-Referentiality

$$\odot\text{–}Unfolding$$

Possible geometry.

Space of the possible.

Stable, closed possibility.

Relation arises from potential.

---

0 – Boundary / Mirroring

The difference encounters itself.

It becomes measurable.

It forms a cut and mirror axis.

$$\boxdot\text{–}Complementarity$$

Impossible geometry.

Space of the impossible.

Unstable, drifting, or not-yet-locked relation.

Complete Process

$$\odot \rightarrow 0 \rightarrow \boxdot$$

Spiral-shaped movement through unfolding, mirroring, and testing.

$$Locking-in : [\boxdot \to |_{=} \to \odot']$$

The new closed possibility carries the traversed trace.

IV. Modes of Difference

|_+ – Expansion

Addition.

$$x|_{+}y$$

x expands by y.

A relation is opened by a further relation.

|_- – Contraction

Subtraction.

$$x|_{-}y$$

x contracts by y.

A relation is led back to a narrower position.

$$|_{\times}\text{–}Resonance$$

Multiplication.

$$x|_{\times}y$$

x resonates with y.

Relations cross planes or rhythms.

Multiplication is not merely repeated addition.

$$|_{\div}\text{–}Disentanglement$$

Division.

$$x|_{\div}y$$

x is disentangled by y.

A total figure is read in recurring relations.

|^ – Exponentiation

Iterated iteration.

$$x|^{y}$$

A relational effect itself becomes the carrier of its repetition.

$$|_{\sqrt{}}\text{–}Rooting$$

Inversion or read-back of exponentiation.

$$|_{\sqrt{}}$$

reads an iterated relation back to its supporting repetition figure.

|_= – Equilibrium

$$Locking-in.$$

Stabilization.

$$x|_{=}y$$

The relation of x and y reaches a stable position under a certain reading.

Equality is not an absolute identity of things.

$$|_{\neq}\text{–}Distinction/Non-coincidence[x|_{\neq}y]$$

x and y do not lock in under the given reading.

Non-coincidence opens the space of impossibility.

It generates no positive object.

|_. – Scale flip

Decimal point.

Change of level or shell.

$$x|.y$$

The reading tilts into another scale direction.

$$|_{\Delta}\text{–}Differential$$

Smallest self-movement.

Local change.

$$|_{\Delta}$$

reads the difference of the change in the immediate beat.

$$|\Sigma|_{\Delta}\text{–}Integral$$

Accumulated self-movement.

$$|\Sigma|_{\Delta}$$

reads the integrated difference along a path.

V. Spaces and Geometries

$$\odot\text{–}PossibleGeometry$$

Stable and closed possibility space.

Readable-back closure.

$$\boxdot\text{–}ImpossibleGeometry$$

Unstable or drifting impossibility space.

Relations that do not yet stabilize.

$$\nabla\text{–}ImpossibleDelta$$

A delta that cannot stabilize under the current reading.

$$\nabla$$

designates no second kind of delta.

It designates \Delta in an impossible or non-locked-in position.

$$|_{\nabla}\text{–}ImpossibleDistinction$$

Distinction in the mode of impossibility.

$$x|_{\nabla}y$$

The relation remains unstable, open, or drifting.

VI. Levels of Distinction

$$\Delta_0 – ZerothLevel$$

Immediate beat.

Ones shell.

$$\Delta_1 – FirstLevel$$

First mirroring.

Tens shell.

\Delta_2 – Second Level Meta-relation

Hundreds shell.

$$\Delta_x – X-thLevel$$

Any further reading depth.

Three-level structure of a number

$$\Delta_2|\Delta_1|\Delta_0$$

The levels are not different kinds of numbers.

They are relational positions of the same number marks.

VII. The Constitution of Numbers

The complete mark space

$$M := 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

There are only the number marks 0 through 9.

All multi-digit numbers are fractal relations of the same ten marks.

0

Closed possibility.

Cutting and mirror axis.

1

First marking.

First directed relation.

2

Mirror relation.

Axis and polarity.

3

Orientation.

First directed closure.

4

Context stabilization.

Order of four.

5

Self-similarity opening.

Pentagon relation.

6

Closed division resonance.

Hexagon relation.

7

Drifting closure.

Heptagon relation.

8

Double folding of four.

Octagon relation.

9

Fracture edge.

Cycle closure.

Transition to the next reading shell.

$$9 := \kappa$$

10

New reading shell.

$$10 := 1|0$$

Ten is not an eleventh basic number.

Multi-digit numbers

$$125 := 1|2|5][2026 := 2|0|2|6$$

A multi-digit number is an ordered fractal inscription.

The place is not a container.

It is relational depth.

VIII. Relation Notations

Basic relation

$$x|y$$

x and y in relation.

Expansion

$$x|_{+}y$$

x expands by y.

Contraction

$$x|_{-}y$$

x contracts by y.

Resonance

$$x|_{\times}y$$

x resonates with y.

Disentanglement

$$x|_{\div}y$$

x is disentangled by y.

Equilibrium

$$x|_{=}y$$

x locks in at y.

$$Non-coincidence[x|_{\neq}y]$$

x differs from y.

The relation remains open or impossible.

Relation triple

$$(a|b|c)$$

Nested relation or delta addressing.

b designates the relational transfer between a and c.

Directions of view

$$-\Delta|+\Delta$$

Negative and positive reading direction.

Number axis

$$(-1|0|1)$$

Complete axial relation of negative direction of view, zero, and positive direction of view.

Negative numbers do not form a second number world.

IX. Geometric Notations

Radius

$$0|1$$

Radius relation in the circle.

Diameter

$$0|2$$

Diameter and axis.

Triangle

$$0|3$$

Triangle in the circle.

Orientation.

Square

$$0|4$$

Square in the circle.

Stable order of four.

\pi becomes readable as a circle-polygon relation.

Pentagon

$$0|5$$

Pentagon in the circle.

\varphi becomes readable as a self-similarity relation.

Hexagon

$$0|6$$

Closed division resonance.

Heptagon

$$0|7$$

Drifting division relation.

Octagon

$$0|8$$

Double folding of four.

Nonagon

$$0|9$$

Fracture edge of the current shell.

X. Special Constants

$$\Delta = 2{,}25^\circ$$

Delta angle.

Minimal self-rotation.

$$\varphi = 1{,}618\ldots$$

Golden ratio.

Self-similarity ratio.

Becomes readable in the pentagon relation.

$$\pi = 3{,}14159\ldots$$

Pi.

Relation of circle and polygon or of closed and linear reading.

Becomes operatively visible in the order of four.

39{,}19

Self-reference constant.

Self-referential constant of the number relation.

$$137{,}5°$$

Golden angle.

Rotational relation of the Fibonacci spiral.

---

e

Euler constant.

Stable self-reference of the rate of change.

$$|_{\Delta} e^{x} = e^{x}$$

XI. Rotation

i

Positive quarter-rotation.

$$i := R_{90°}$$

-i

Inverse quarter-rotation.

$$-i := R_{-90°}$$

Rotation cycle

$$i^2 := R_{180°}][i^3 := R_{270°}][i^4 := id$$

i is not an additional kind of number.

i is a rotation address of the same relational figure.

XII. Frame and Delta Configuration

|_| – Frozen Frame

Closed local possibility.

$$F := |_|$$

$$|_{\Delta} - MarkedFrame$$

Frame with set Delta marking.

Direction set

$$D := N, E, S, W$$

Delta configuration

$$o : D \to 0, 1$$

o designates the set and open Delta positions around the frame.

Number of set positions

$$k(o) := \sum_{d \in D} o(d)$$

0Δ

Pure frame.

No set Delta position.

1Δ

Event.

One set Delta position.

2Δ

Axis.

Two set Delta positions.

3Δ

Observer orientation.

Three set positions and one open direction.

4Δ

Full context space.

Four set positions.

$$4\Delta = \Delta^{\Delta}$$

---

ho – Hole direction

Open direction of a three-Delta state.

Epistemic outside of the observer.

$$\mathrm{ori}(o)$$

Orientation reading of a three-Delta state.

$$\mathcal{O}$$

Set of all orientations.

$$\mathcal{O} = o_N, o_E, o_S, o_W$$

$$\mathcal{T}$$

Transition space of the orientations.

$$|\mathcal{T}| = 12$$

Twelve directed paths between the four observer positions.

$$o_{\mathrm{sat}}$$

Saturated four-Delta configuration.

Full context space.

XIII. Observer Notations

$$\circledcirc_B$$

Closed observer reading of B.

$$\mathcal{O}_B$$

Orientation state of observer B.

$$(\circledcirc_B | P)$$

Relation between observer B and phenomenon P.

Observer and phenomenon arise within their relation.

XIV. Fractal Operators

$$\ominus\text{–}FractalDifference$$

Difference between orders of difference.

$$\circledast\text{–}FractalLinkage$$

Binding of recursive orders of difference.

$$\mathcal{R}\text{–}Rotation$$

Rotation of an orientation or reading position.

$$\tilde{\mathcal{O}}\text{–}OrientationOscillator$$

Recurring change of orientation states.

$$\mathcal{F}\text{–}PhaseFeedback$$

A reading changes the condition of its next reading.

$$\zeta_\Delta(s)\text{–}Zeta - DeltaCondensation$$

Condensation of Delta traces under the reading position s.

Scale-spanning read-back of orders of repetition.

XV. Inscription and Path

I – Inscription

$$I = (m_0, m_1, \ldots, m_{N-1})$$

Ordered sequence of visible marks and operator edges.

$$\gamma_I – Path of the inscription [\gamma_I = (e_0, e_1, \ldots, e_{N-1})]$$

Clocked sequence of the operative transitions of an inscription.

---

e_n – Edge

$$e_n : x_n \to x_{n+1}$$

An operative step within the path.

T_n – Clocked reading operation

$$T_n := Step \circ O_n$$

Non-commutativity [Step\circ O\neq O\circ Step]

The order of step and reading carries meaning.

XVI. Trace and Visibility

we – Difference reading of an edge

Relational contribution of an edge to the trace.

$$\chi_{\text{vis}} – Visibility rejector [\chi_{\text{vis}} \in 0, 1]$$

Determines whether an edge is visible under the current reading.

$$\chi_{\text{gate}} – Gate rejector [\chi_{\text{gate}} \in 0, 1]$$

Determines whether an edge is admitted to the current enactment.

H_n – Step contribution

$$H_n := \chi_{\text{vis}}(e_n) \cdot \chi_{\text{gate}}(e_n) \cdot w(e_n)$$

$$\Sigma(\gamma) – Trace [\Sigma(\gamma) := \sum_{e \in \gamma} w(e)]$$

Integrated difference of a path.

Trace accumulation

$$\Sigma_{n+1} = \Sigma_n + H_n$$

L_n – Readable edges

Set of the edges visible or admitted, as applicable, at beat n.

XVII. Holonomy

$$\Omega(n) – Accumulated holonomy [\Omega(n) := \sum_{t=0}^{n-1} H(t)]$$

$$\phi(n) – Phase position [\phi(n) := \Omega(n) \bmod 2\pi]$$

Non-vanishing return difference

$$\gamma(0) = \gamma(1)$$

and:

$$\Sigma(\gamma) \neq 0$$

Return to the starting place without return to the starting position.

$$\gamma_\infty – Infinity loop$$

Closed register form in which drift remains bound in circulation.

XVIII. Projection and Binding

$$\Pi_i – Projection [\Pi_i : \Sigma \to A_i]$$

Reading channel of a trace.

---

a_i – Projected readout

$$a_i := \Pi_i(\Sigma)$$

$$\Delta a_i – Difference of a reading channel [\Delta a_i(n) := a_i(n) | a_i(n+1)]$$

$$\bowtie – Binding gate$$

Checks the compatibility of different trace readings.

$$\Delta a_i \bowtie \Delta a_j$$

XIX. The ¾Corner

$$\Theta_{\frac{3}{4}} – ¾ Corner [\Theta_{\frac{3}{4}} := (e_1, e_2, e_3; D)]$$

Three set hinge edges and one open drift edge.

$$\mathcal{L}_{\frac{3}{4}} – Rattle space$$

Finite space of possible snapping positions.

L^* – Minimal drift position

$$L^* := \arg\min_{L \in \mathcal{L}_{\frac{3}{4}}} drift(L)$$

$$Lock_{\frac{3}{4}} – Equality snap$$

Finite snapping-in of a relation in its minimal drift position.

XX. Normal Form and Reading

$$NF(\Sigma) – Normal form$$

Self-resonance reading of a trace.

Stable position that can be read back.

$$read(\Sigma) – Public reading$$

Visible readout of a trace.

$$read_L$$

Left-side reading of an equation inscription.

$$read_R$$

Right-side reading of an equation inscription.

$$\equiv_\Sigma \text{–} Trace - classequality$$

Back-addressable equality of different readings.

$$G \equiv_\Sigma I$$

G and I carry the same trace class.

$$\sim \text{–} Resonancecomparison$$

Resonance similarity without classical equality.

$$\rightsquigarrow \text{–} Readingtransition$$

A form falls under a reading onto another visible form.

---

recover – Read-back

Reconstruction of a trace from its readout.

Validity

$$valid(L) \iff recover(L) \equiv_\Sigma \Sigma$$

A label is valid if it remains readable back to its trace class.

XXI. Complete Enactment of Reading

$$I \to \gamma_I \to \Sigma(\gamma_I) \to \Pi_i(\Sigma) \to \bowtie \to Lock_{\frac{3}{4}} \to NF \to read$$

The inscription is read as a path.

The path carries a trace.

The trace is projected.

The projections are bound.

The drift snaps in.

The trace forms a normal form.

The normal form is read publicly.

XXII.

Isomorphy of Reading

In the Tractatus Principio, isomorphy designates neither the equality of different signs nor their free interchangeability. It designates the preservation of the same relational order in the transition from one reading form into another. Difference, Delta, number, geometry, observer, and trace are therefore not identical; they form different operative modes of appearance of a structure whose transitions, orientations, and back-references remain mappable onto one another. Two readings are isomorphic if one can be translated into the other and then read back to the same trace order, without the supporting difference structure being lost. Isomorphy thus means: The visible form changes, while the order of its relations is preserved.

$$\boxed{L_1 \cong L_2 \iff recover(L_1) \equiv_{\Sigma} recover(L_2)}$$

The isomorphy therefore lies not in the signs themselves, but in the read-back capability of their relational constitution.

Part II

From the Sign to the Executable Sign

Python as Representation of Difference

Foreword

The second writing

The first part of this book has brought the signs of the Tractatus into an operative sequence:

$$|\rightarrow \Delta \rightarrow Number \rightarrow Operation \rightarrow Geometry \rightarrow Observer \rightarrow Trace \rightarrow Reading$$

This sequence does not end with a formula.

It can be executed.

For this, no new theory has to be introduced. It only has to be shown how the same order of difference appears in a programming language.

The Tractatus notation writes the relation in condensed form.

Python unfolds the same relation as an ordered enactment.

Tractatus notation ≅ Python code

This isomorphy does not mean equality of the signs.

The stroke

$$|$$

is not a Python object.

Delta

$$\Delta$$

is not a function.

The trace

$$\Sigma$$

is not an ordinary list.

The readout

$$read(\Sigma)$$

is not an arbitrary output.

Python represents these forms by arranging their relations, transitions, and reading conditions in executable writing.

The code therefore does not deal with objects.

It writes conditions of distinguishability.

It writes what may be bound.

It writes what becomes visible in a beat.

It writes which trace an enactment leaves behind.

It writes how an address becomes readable.

It writes under which conditions an opening can be closed.

The code does not produce the result as a finished object.

It makes the difference readable.

Chapter X

Code as Inscription

10.1 Code is not the difference

The difference is primitive:

$$|$$

Python is not primitive.

Python is a public script in which relations can be marked, ordered, and executed.

Therefore:

$$\text{Python code} \neq |$$

and at the same time:

$$\text{Python code} \text{executable inscription of } |$$

A program does not contain the difference like an object.

It establishes an order in which differences can become visible.

10.2 The ordinary reading of code

An ordinary program is often read in the form:

$$Input \rightarrow Algorithm \rightarrow Output$$

For example:

```
def add(a, b):
    return a + b
```

This reading presupposes:

- a and b are values,
- + is a calculation function,
- the function generates a new value,
- return outputs this value.

The difference appears in this only as the already presupposed separation of values and operations.

It is not itself written.

10.3 The Tractatus reading of code

Delta-constraint code reads the same process differently:

$$Input \to question figure \to Gate \to trace \to holonomy \to address \to Label$$

The code does not first ask:

$$\textbf{What value should I generate?}$$

It asks:

$$\textbf{Which relation is present?][Which opening does this relation carry?][Which transitions become visible?][Which trace remains behind?][Which address closes the opening?][Which label is read at this address?}$$

The elementary sequence is therefore:

$$a|_\mu b \to \Sigma \to Addr \to read(Addr)$$

In Python:

```
binding = bind(a, b, mode)
trace = integrate(binding)
address = locate(trace)
label = read(address)
```

These four lines do not designate four mutually independent machines.

They unfold a single relation in four reading positions:

$$\textbf{Binding} \to \textbf{trace} \to \textbf{address} \to \textbf{public reading}$$

10.4 Code as ordered inscription

A line of code is not a mere instruction.

It is a marked position within a sequence.

```
binding = bind(a, b, mode)
```

This line marks a binding.

```
trace = integrate(binding)
```

This line marks the inscription of the enactment.

```
label = read(trace)
```

This line marks the public reading.

The meaning does not lie in the lines by themselves.

It arises from their order.

$$\mathit{Step} \circ O \neq O \circ \mathit{Step}$$

Correspondingly, these two programs are also not necessarily the same:

```
trace = integrate(binding)
label = read(trace)
```

and:

```
label = read(binding)
trace = integrate(label)
```

The second program changes the order of the reading.

It reads before a trace has arisen.

The order is not only technical syntax.

It is memory.

Chapter XI

The three layers of code

11.1 The necessity of separation

The code of difference must keep three layers apart:

$$\mathcal{C} \neq \mathcal{I} \neq \mathcal{L}$$

Here:

$$\mathcal{C}\text{Carrier layer}][\mathcal{I}\text{nscription layer}][\mathcal{L}\text{interpretation layer}$$

The separation prevents a visible mark from being confused with its trace, or a trace with its binding structure.

11.2 Carrier layer

The carrier layer carries the binding:

$$a|_{\mu}b$$

Here lie:

- poles,
- openings,
- coupling modes,
- directions,
- relational positions.

In Python, such a binding can be represented as:

```
binding = Binding(
    left=a,
    mode=mode,
    right=b,
)
```

This form does not say that a and b are two finished things.

It marks two positions within a binding.

mode, too, is not an object between them.

mode designates how the binding is read.

11.3 Inscription layer

The inscription layer carries the enactment:

$$\gamma, \quad H_n, \quad \Sigma, \quad \Omega, \quad Addr$$

In Python:

```
path = trace_path(binding)
trace = integrate(path)
address = locate(trace)
```

No number object arises here.

Here the order of transitions is inscribed.

The trace is the memory of this order.

11.4 Interpretation layer

The interpretation layer carries visible labels:

$$0, 1, 2, \ldots, 9, \quad x, \quad +, \quad =, \quad Ans$$

In Python:

```
label = read(address)
```

The label is not the address.

$$Label \neq Addr$$

It is the public reading of the address:

$$Label := read(Addr)$$

A digit is therefore not a kernel value.

An operator sign is not a kernel operation.

A variable is not a container.

A display is not the enactment.

11.5 The forbidden mixing

Invalid is:

```
digit = address
```

For:

$$digit \neq address$$

Valid is:

```
digit = read(address)
```

Invalid is:

```
operator = numeric_function
```

Valid is:

```
operator = binding_mode
```

Invalid is:

```
answer = trace
```

Valid is:

```
answer = read(trace)
```

The difference between the two forms is not stylistic.

It preserves the layers of the reading.

Chapter XII

The stroke in code

12.1 Difference of the first order

Tractatus:

The difference of the first order does not yet possess determinate poles.

It is the condition for poles to be able to appear.

Python cannot contain this condition as a finished object.

It can only mark a place at which a distinction may be enacted.

For example:

```
def distinguish(left, right):
    return left, right
```

This function is not yet a sufficient representation.

It merely returns two elements.

But the difference does not lie in the elements.

It lies in their binding.

Therefore the relation itself must be marked:

```
def distinguish(left, right):
    return Binding(left=left, right=right)
```

The class Binding is still not the difference itself.

It is its Python inscription.

12.2 Difference of the second order

Tractatus:

$$A|B$$

Python:

```
relation = Binding(
    left=A,
    right=B,
)
```

A and B receive their position only through the binding.

Without the binding they are not poles of this relation.

Therefore the ordinary reading:

```
A = object()
B = object()
```

is not fundamental for the kernel.

Fundamental is:

```
relation = distinguish(A, B)
```

The poles do not generate the relation.

The relation distinguishes the poles.

12.3 Difference of the third order

Tractatus:

$$(A|B)|(C|D)$$

Python:

```
left_relation = distinguish(A, B)
right_relation = distinguish(C, D)

relation = distinguish(
    left_relation,
    right_relation,
)
```

Here, four objects are not being collected.

Two distinctions already enacted are distinguished again.

The relational depth increases.

12.4 Self-referentiality

Tractatus:

$$|(|)$$

Python:

```
first_difference = distinguish(A, B)

self_related_difference = distinguish(
    first_difference,
    first_difference,
)
```

This representation is self-referential only if the earlier relation remains part of the new relation.

A mere repetition is not enough:

distinguish(A, B)

distinguish(A, B)

Here there are only two identical calls one after the other.

Self-referentiality arises only when a relation acts back upon its own relational form.

$$\Delta = |(|)$$

Python:

```
delta = relate(
    relation,
    to=relation,
)
```

delta is thereby not a thing named Delta.

It designates the marked efficacy of self-relation.

Chapter XIII

Delta as transition

13.1 Delta is not a value

Tractatus:

$$A|B;\Delta;C|D$$

Delta designates the transition of one difference position into another.

It is not the initial state.

It is not the final state.

It is also not a third object between the two.

Python:

```
next_relation = transition(current_relation)
```

The function transition represents the transition.

It is not Delta itself.

Delta lies in the effective relation of initial position, transition, and subsequent position.

13.2 Delta as beat

Tractatus:

$$o_n \to o_{n+1}$$

Python:

```
for tick in transitions:
    orientation = tick.apply(orientation)
```

The beat is not only a counting variable.

It designates the ordered place of the transition.

```
for tick_index, transition in enumerate(path):
    state = transition.apply(state)
```

tick_index is the visible marking of the beat.

The beat itself lies in the enactment:

$$state_n \rightarrow state_{n+1}$$

13.3 Delta as drift

Tractatus:

$$\Delta\varphi$$

Python:

```
drift = read_drift(
    before=orientation_a,
    after=orientation_b,
)
```

The drift is not read as a property of an individual state.

It appears between two reading positions.

Therefore this would be imprecise:

state.drift

Stricter is:

```
drift = difference_between(
    state_a,
    state_b,
)
```

The drift belongs to the relation.

13.4 Delta as self-rotation

Tractatus:

$$\Delta = 2{,}25^\circ$$

Python can mark this self-rotation as the smallest permitted rotation interval:

```
DELTA_ROTATION = RotationStep(
    numerator=225,
    denominator=100,
)
```

What is decisive is not the data type.

What is decisive is that the rotation is read as a transition between orientations:

```
next_orientation = rotate(
    orientation,
    by=DELTA_ROTATION,
)
```

The number 2.25 alone is not yet a self-rotation.

Only its position in the enactment of rotation makes it a Delta address.

Chapter XIV

Number and mark

14.1 Number as readout

Tractatus:

$$n := read(\Sigma_n)$$

Python:

```
digit = read(trace)
```

The visible mark appears at the end of the reading.

Not:

```
trace = digit
```

but:

```
digit = read(trace)
```

Thus the number is not an original unit in the kernel.

It is the public reading of a trace.

14.2 The marks from zero to nine

Tractatus:

$$M = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$$

Python:

```
MARKS = ("0", "1", "2", "3", "4",
         "5", "6", "7", "8", "9")
```

The quotation marks are important.

They show that the visible digits are initially marks.

"5"

is a visible inscription.

5

is already a numerical value in Python.

For the interpretation layer, the mark is therefore initially more precise:

```
mark = "5"
```

Only an explicit reading rule may connect it with an address:

```
mark = read(address)
```

14.3 Multi-digitness

Tractatus:

$$125 := 1|2|5$$

Python:

```
inscription = ("1", "2", "5")
```

The sequence is initially not a number entity.

It is an ordered inscription.

```
for mark in inscription:
    trace = trace_mark(mark, trace)
```

Thus:

[125

$$read(1 \to 2 \to 5)]$$

Multi-digitness arises through relational depth.

14.4 The place

Tractatus:

$$\Delta_2|\Delta_1|\Delta_0$$

Python:

```
positions = (
    Position(depth=2, mark="1"),
    Position(depth=1, mark="2"),
    Position(depth=0, mark="5"),
)
```

The position is not a container in which the digit lies.

It designates the reading depth of the mark within the inscription.

14.5 The nine as fracture edge

Tractatus:

$$9 := \kappa$$

Python:

```
KAPPA = Mark(
    label="9",
    role="shell_break",
)
```

The mark 9 is not read merely as another entry in a series.

Its role marks the closure and transition of the basic shell.

14.6 Ten as new shell

Tractatus:

$$10 := 1|0$$

Python:

```
ten_inscription = (
    Mark("1"),
    Mark("0"),
)
```

Not:

```
NEW_DIGIT = "10"
```

For ten is not an eleventh basic mark.

It is the first visible relation of a new reading shell.

Chapter XV

Operators as binding modes

15.1 Operators do not generate values

Tractatus:

$$a|_{+}b$$

Python:

```
binding = bind(a, b, mode=EXPANSION)
```

The operator + is read as a mode of expansion.

Not:

```
result = a + b
```

In ordinary Python, this line immediately generates a value.

The difference-logical representation keeps the relation open:

```
binding = Binding(left=a, mode=EXPANSION, right=b)
```

15.2 Expansion

Tractatus:

|+

Python:

```
EXPANSION = Mode("expansion")
```

Application:

```
relation = bind(
    left=a,
    right=b,
    mode=EXPANSION,
)
```

Expansion does not automatically mean numerical addition.

It designates the direction in which two reading positions enter into an expanded relation.

15.3 Contraction

Tractatus:

|–

Python:

```
CONTRACTION = Mode("contraction")
```

Application:

```
relation = bind(
    left=a,
    right=b,
    mode=CONTRACTION,
)
```

Contraction does not simply delete a value.

It orients the relation toward withdrawal, retraction, or diminution of its visible binding.

15.4 Resonance

Tractatus:

$$|\times$$

Python:

```
RESONANCE = Mode("resonance")
```

Application:

```
relation = bind(
    left=a,
    right=b,
    mode=RESONANCE,
)
```

Multiplication thereby appears as a coupling of rhythms or recurring bindings.

It is not originally the product of two number objects.

15.5 Disentanglement

Tractatus:

$$|\div$$

Python:

```
DISENTANGLEMENT = Mode("disentanglement")
```

Application:

```
relation = bind(
    left=a,
    right=b,
    mode=DISENTANGLEMENT,
)
```

Division appears as the question of how a bound relation can be read into distinguishable partial relations.

15.6 Equilibrium

Tractatus:

$$|=$$

Python:

```
EQUILIBRIUM = Mode("equilibrium")
```

The equality sign does not compare two finished objects.

It activates the test of whether different readings lock onto the same trace class.

```
equation = bind(
    left=left_inscription,
    right=right_inscription,
    mode=EQUILIBRIUM,
)
```

After that:

```
valid = same_trace_class(
    read_left(equation),
    read_right(equation),
)
```

Equality is read as trace equivalence:

$$\Sigma_A \equiv_\Sigma \Sigma_B$$

15.7 Lowering

An ordinary expression:

a + b

is lowered into the carrier layer:

$$a + b \mapsto a|\mu{+}b$$

Python:

```
binding = lower(
    left=a,
    operator="+",
    right=b,
)
```

Result of the lowering:

Binding(

```
    left=a,
    mode=EXPANSION,
    right=b,
)
```

Lowering does not calculate.

It translates a visible operator sign into a binding mode.

Chapter XVI

Variable and Opening

16.1 The variable is not a container

Ordinary reading:

```
x = 5
```

x contains the value 5.

Difference-logical reading:

x marks an addressed place at which the mark 5 is readable under a certain relation.

The assignment is thus a binding:

$$x|_{=}!5$$

Python:

```
assignment = bind(
    left=Opening("x"),
    right=Mark("5"),
    mode=EQUILIBRIUM,
)
```

16.2 The opening

Tractatus:

$$u_j := 0_{R_Q}$$

The opening is not an absolute emptiness.

It is a relative zero in the relation of the question.

Python:

```
opening = Opening(
    name="u1",
    relative_to=question_relation,
)
```

The opening contains no missing value.

It marks a role not yet closed.

16.3 The question figure

Tractatus:

$$F_Q = (G_Q, U_Q, R_Q)$$

Python:

```
question = QuestionFigure(
    graph=binding_graph,
    openings=(opening,),
    relation=question_relation,
)
```

Here:

question.graph

is the binding structure,

question.openings

is the set of openings,

and:

question.relation

is the relation under which the openings must be closed.

16.4 Implicit opening

The visible inscription:

$$2 + 2$$

becomes:

$$2|\mu+2|\mu =u_1$$

Python:

```
expression = bind(
    Mark("2"),
    Mark("2"),
    mode=EXPANSION,
)

question = bind(
    expression,
    Opening("u1"),
    mode=EQUILIBRIUM,
)
```

The question is not:

What numerical value does the addition produce?

It is:

Which address closes the opening of this expansive equilibrium figure?

16.5 Explicit opening

The inscription:

$$x + 3 = 7$$

is read as:

$$u_1|\mu+3|\mu =7$$

Python:

```
opening = Opening("u1")

left_side = bind(
    opening,
    Mark("3"),
    mode=EXPANSION,
)

question = bind(
    left_side,
    Mark("7"),
    mode=EQUILIBRIUM,
)
```

x is not an unknown number.

x is the visible label of an open relational place.

16.6 Closure

Tractatus:

$$close(u_j, Addr_j) = 1$$

Python:

```
closed = opening.close_with(address)
```

The address is not inserted into a container.

It snaps into the relation of the opening.

Validity:

```
closed = compatible(
    opening,
    address,
    question_relation,
)
```

Chapter XVII

Gate and Visibility

17.1 The gate does not decide truth

Tractatus:

$$\chi_{\text{vis}}(e_n)$$

Python:

if visible(edge, tick):

```
trace.append(edge)
```

The gate does not say:

The edge is true.

It says:

The edge contributes to the visible trace in this tick.

17.2 Visibility

Tractatus:

$$L_n$$

Python:

```
visible_edges = visibility_mask(
    path,
    tick=n,
)
```

L_n is the reading mask of the tick.

It determines which transitions are read effectively in this tick.

17.3 Gate and contribution

Tractatus:

$$H_n := \chi_{\text{vis}}(e_n)\chi_{\text{gate}}(e_n)w(e_n)$$

Python:

if visible(edge, tick) and allowed(edge, tick):

```
    contribution = read_difference(edge)
else:
    contribution = no_contribution()
```

The two conditions are rejectors.

They do not generate any additional value.

They allow a difference to contribute to the trace or not.

17.4 Condition in Python code

Ordinarily:

if condition:

```
do_something()
```

$$Difference - logically:$$

if gate.is_open(relation, tick):

```
trace.inscribe(relation)
```

The if becomes the gate.

It does not separate two finished worlds into true and false.

It marks the visibility of a relation in the current tick.

17.5 The closed gate

Tractatus:

$$\chi_{\text{gate}} = 0 \Rightarrow H_n = 0$$

Python:

if not gate.is_open(edge, tick):

```
return no_contribution()
```

No trace does not mean that the relation has ontologically disappeared.

Under this reading condition, it does not contribute to the visible inscription.

Chapter XVIII

Path, Loop, and Trace

18.1 The inscription as path

Tractatus:

$$I \to \gamma_I$$

Python:

```
path = trace_path(inscription)
```

An inscription is not treated as a static syntax tree.

It is read as an ordered sequence of transitions.

18.2 The loop

Ordinarily:

```
for edge in path:
    process(edge)
```

$$Difference - logically:$$

```
for tick, edge in enumerate(path):
    contribution = read_step(edge, tick)
    trace.inscribe(contribution)
```

The loop does not merely repeat a command.

It carries the relation through a sequence of ticks.

Each pass has a different position in the overall enactment.

18.3 The trace

Tractatus:

$$\Sigma_{n+1} = \Sigma_n + H_n$$

Python:

trace.append(contribution)

The method append is not merely a technical list operation here.

It shows:

The new contribution does not replace the earlier trace.

It inscribes itself into it.

18.4 Trace as the only memory

A state machine asks:

$$current_state$$

The delta-constraint code asks:

$$trace_so_far$$

For the meaning lies not only in the present state.

It lies in the path by which this state was reached.

Two equal states can carry different traces.

```
same_state = state_a == state_b
different_trace = trace_a != trace_b
```

Both can hold at the same time.

18.5 Holonomy

Tractatus:

$$\gamma(0) = \gamma(1)$$

and yet:

$$\Sigma(\gamma) \neq 0$$

Python:

```
same_position = path.start == path.end
remaining_trace = trace.is_nonzero()
```

Complete:

if same_position and remaining_trace:

```
holonomy = read_return_difference(trace)
```

The return to the same place is not a return to the same situation.

18.6 Non-commutativity

Tractatus:

$$(a|\mu b)|\nu c \neq a|\mu(b|\nu c)$$

Python:

```
first = bind(
    bind(a, b, mode=mu),
    c,
    mode=nu,
)
```

```
second = bind(
    a,
    bind(b, c, mode=nu),
    mode=mu,
)
```

first and second carry different paths.

An optimization must not exchange them without an explicit rule.

Invalid:

```
expression = optimizer.reassociate(expression)
```

Valid only with an indicated rule:

if rules.allow_association(mu, nu):

```
expression = reassociate(expression)
```

The order is trace.

Chapter XIX

Observer and Context in Code

19.1 The frame

Tractatus:

$$F := |_|$$

Python:

```
frame = Frame()
```

Frame() is not an empty container.

It denotes the closed reading frame within which markings can occur.

19.2 Delta configuration

Tractatus:

$$o : D \to 0,1$$

Python:

```
orientation = {
    "N": 1,
    "E": 1,
    "S": 1,
    "W": 0,
}
```

The entries denote visible or unoccupied directional places.

19.3 Observer

Tractatus:

$$3\Delta = 4\Delta - \Delta$$

Python:

```
observer = context.with_missing(
    direction="W",
)
```

The observer is not inserted into the context as an additional object.

It appears as a full context with directed incompleteness.

19.4 Hole direction

Tractatus:

$$h(o)$$

Python:

```
hole_direction = observer.missing_direction()
```

The hole direction is not a defect of the program.

It is the orientation of the observer.

19.5 Full context

Tractatus:

$$4\Delta = \Delta^{\Delta}$$

Python:

```
context = DeltaContext(
    north=True,
    east=True,
    south=True,
    west=True,
)
```

The full context contains no preferred reading direction.

Only the removal of a marking generates orientation:

```
observer = context.with_missing("N")
```

19.6 Observation as projection

The observer does not necessarily change the carrier.

It reads it under an orientation.

Tractatus:

$$\Pi_o(\Sigma)$$

Python:

```
observed_trace = project(
    trace,
    orientation=observer,
)
```

The projection should remain read-only.

Invalid:

observer.project(trace).mutate_kernel()

Valid:

```
view = observer.read(trace)
```

The observer does not generate the trace.

It makes one of its readings visible.

Chapter XX

Projection, Binding, and Equality

20.1 One trace, several readings

Tractatus:

$$a_i = \Pi_i(\Sigma)$$

Python:

```
visible_view = project_visible(trace)
rotation_view = project_rotation(trace)
operator_view = project_operator(trace)
drift_view = project_drift(trace)
```

These projections are not separate states.

They are different reading channels of the same trace.

20.2 Read-only projection

Python:

```
class Projection:
    def read(self, trace):
        return view_of(trace)
```

Not:

```
class Projection:
    def transform_kernel(self, trace):
        ...
```

The projection reads.

It does not rewrite the kernel.

20.3 Binding gate

Tractatus:

$$\Delta a_i \bowtie \Delta a_j$$

Python:

```
compatible = bindable(
    projection_a,
    projection_b,
)
```

The projections do not have to be identical.

They must fit together under the question relation.

20.4 Validity

Tractatus:

$$valid(I) \iff \forall i, j : \Pi_i(\Sigma(I)) \bowtie \Pi_j(\Sigma(I))$$

Python:

```
valid = all(
    bindable(left, right)
    for left in projections
    for right in projections
)
```

An invalid inscription does not necessarily have an empty value.

It has no bindable commit form.

20.5 Equality as measuring point

Tractatus:

$$= := \Pi + \bowtie + Lock_{\frac{3}{4}} + NF$$

Python:

```
def equality_reading(inscription):
    trace = integrate(inscription)
    projections = project(trace)
    bound = bind_projections(projections)
    locked = lock_three_quarters(bound)
    return normal_form(locked)
```

The equality sign thereby does not become:

```
left == right
```

but rather a sequence of readings, bindings, and lockings.

Chapter XXI

Normal Form and Readout

21.1 Normal form is not simplification

Tractatus:

$$B \to B_{NF}$$

Python:

```
normal = recognize_form(binding)
```

Not:

```
normal = calculate_value(binding)
```

The normal form recognizes the type of the binding structure.

21.2 Self-binding

Tractatus:

$$a|\mu b|\nu a \to \Delta_{\mu,\nu}(a,b)$$

Python:

if pattern.is_self_binding(binding):

```
normal = DeltaPattern.from_binding(binding)
```

This transformation calculates no value.

It recognizes a recurring relational form.

21.3 No hidden simplification

A repetition must not automatically disappear:

bind(a, a, mode)

may collapse to a only if the mode is explicitly stabilizing:

if rules.is_stabilizing(mode):

```
return a
```

Without a rule, the repetition remains part of the trace.

21.4 Address

Tractatus:

$$Addr_i := read(\Omega_i | D_i^{out} | R_Q)$$

Python:

```
address = read_address(
    holonomy=omega,
    outer_direction=outer_direction,
    question_relation=question_relation,
)
```

The address is not a visible number.

It designates a position under which a trace can be read.

21.5 Readout

Tractatus:

$$Ans := read(Addr)$$

Python:

```
answer = read(address)
```

The answer is not generated.

It becomes publicly readable at the address.

21.6 Read-back capability

Tractatus:

$$valid(L) \iff recover(L) \equiv_\Sigma \Sigma$$

Python:

```
recovered_trace = recover(label)

valid = same_trace_class(
    recovered_trace,
    original_trace,
)
```

A label is grounded only if it can be traced back to its trace class.

Invalid:

```
return "5"
```

without an address.

Valid:

```
return read(address)
```

with a read-back-capable trace.

Chapter XXII

Error as relational error

22.1 Error is not necessarily falsehood

In Delta-Constraint-Code, an error does not designate a merely false number.

It designates a disturbance of the reading architecture.

22.2 No question figure

Tractatus:

$$|U_Q| = 0$$

Python:

if not question.openings:

```
raise NotAQuestionError()
```

A closed inscription is not a task.

22.3 Unbound opening

Tractatus:

$$u_j \not\bowtie R_Q$$

Python:

if not bindable(opening, question_relation):

```
raise UnboundOpeningError()
```

22.4 Ungrounded label

Tractatus:

$$\neg \exists Addr : L = read(Addr)$$

Python:

if label.address is None:

```
raise UngroundedLabelError()
```

A visible result without an address is not readable back.

22.5 Hidden calculation

Invalid:

```
result = a + b

if this calculation is presented as a primitive operation in the kernel.
```

Valid:

```
edge = bind(
    a,
    b,
    mode=EXPANSION,
)
```

The classical calculation may appear as a public projection.

It must not become the ontological core unnoticed.

22.6 Constructed holonomy

Invalid:

```
trace = construct_trace_for(
    desired_answer,
)
```

For here the trace is generated from a previously posited result.

Valid:

```
trace = integrate(question_figure)
answer = read(locate(trace))
```

The trace precedes the readout.

22.7 Covert reordering

Invalid:

optimizer.reorder(binding)

without an explicitly indicated relational rule.

Valid:

if rules.allow_reordering(binding):

```
binding = reorder(binding)
```

The order belongs to the enactment.

Chapter XXIII

A complete double reading

23.1 The inscription

We consider:

$$2 + 2$$

The visible inscription is not yet a finished calculation.

It carries an implicit opening.

23.2 Tractatus notation

$$2|\mu+2|\mu = u_1$$

From this the question figure arises:

$$F_Q = (G_Q, u_1, R_Q)$$

The figure is read as a path:

$$F_Q \to \gamma_Q$$

The visible contributions are integrated:

$$\Omega_{n+1} := \Omega_n + H_n$$

The address is read:

[Addr :=

$$read(\Omega|D^{out}|R_Q)]$$

The opening is closed:

$$close(u_1, Addr) = 1$$

The public label appears:

$$Ans = read(Addr)$$

23.3 Python inscription

```
left_mark = Mark("2")
right_mark = Mark("2")

expansion = bind(
    left_mark,
    right_mark,
    mode=EXPANSION,
)

opening = Opening(
    name="u1",
)

question = bind(
```

```
    expansion,
    opening,
    mode=EQUILIBRIUM,
)

path = trace_path(question)

trace = Trace()

for tick, edge in enumerate(path):
    if visible(edge, tick) and allowed(edge, tick):
        contribution = read_difference(
            edge,
            tick,
        )
        trace.append(contribution)

address = read_address(
    trace=trace,
    question=question,
)

answer = read(address)
```

23.4 Reading

Mark("2")

designates a visible mark.

bind(..., mode=EXPANSION)

lowers the plus sign into a binding mode.

Opening("u1")

sets the relative zero of the question.

trace_path(question)

reads the question figure as an ordered path.

visible(...)

and:

allowed(...)

form the gates.

trace.append(contribution)

writes the visible difference into the trace.

read_address(...)

reads the position under which the opening can be closed.

read(address)

projects the public label.

At no point has the code posited a result as an original value.

It has written conditions under which a trace becomes readable as an answer.

Chapter XXIV

Python as second notation

24.1 No competition between the scripts

Tractatus notation and Python do not stand opposed to one another.

The formula condenses.

The code unfolds.

Tractatus:

$$a|_{\mu}b \to \Sigma \to Addr \to read(Addr)$$

Python:

```
binding = bind(a, b, mode)
trace = integrate(binding)
address = locate(trace)
label = read(address)
```

Both inscriptions carry the same order of movement.

24.2 The formula shows the relation

Tractatus notation lets the overall form become visible at a glance.

$$F_Q \to \gamma_Q \to \Omega \to Addr \to Ans$$

It shows the position of the concepts within the whole.

24.3 The code shows the enactment

Python unfolds the same overall form into a temporal sequence:

```
question = form_question(inscription)
path = trace_path(question)
trace = integrate(path)
address = locate(trace)
answer = read(address)
```

The code makes visible which operation must happen before which other one.

24.4 Translation without loss

A valid translation must preserve the order of difference.

It must preserve:

- the position of the poles,
- the binding mode,
- the sequence,
- the openings,
- the gates,
- the trace,
- the address,
- the read-back capability.

Formal:

$$T : \text{Tractatus} \to \text{Python}$$

is valid if:

$$recover(T(I)) \equiv_{\Sigma} I$$

The Python inscription must be able to be traced back to the same trace class.

24.5 The code as application

Application here does not mean that an abstract theory is subsequently transferred onto an external object.

Application means:

$$\textbf{The relation becomes executable.}$$

The difference is not transformed into Python.

Python is ordered so that it can represent the difference.

Chapter XXV

Basic rules of Delta-Constraint-Coding

Rule 1

$$\boxed{|\ \text{is primitive.}}$$

No numerical value, no object, and no function precedes the distinction.

Rule 2

$$\boxed{L := read(\Sigma)}$$

Every visible label is a reading.

Rule 3

$$\boxed{op \rightsquigarrow |\mu op}$$

Every operator is read as a binding mode.

Rule 4

$$\boxed{F_Q = (G_Q, U_Q, R_Q)}$$

A task is a question figure.

Rule 5

$$\boxed{|U_Q| \geq 1}$$

Every question possesses at least one opening.

Rule 6

$$\boxed{u_j := 0_{R_Q}}$$

The opening is relative zero in the relation of the question.

Rule 7

$$\boxed{a\ op\ b \mapsto a|\mu op b}$$

Visible operators are lowered into binding modes.

Rule 8

$$\boxed{\Omega_{n+1} := \Omega_n + H_n}$$

The trace arises through ordered accumulation.

Rule 9

$$\boxed{Addr := read(\Omega | D^{out} | R_Q)}$$

The address is a reading of the holonomy under the question relation.

Rule 10

$$\boxed{Ans := read(Addr)}$$

The answer is the public readout of the closing address.

Rule 11

$$\boxed{recover(Ans) \equiv_\Sigma \Sigma}$$

Every valid answer remains readable back to its trace class.

Rule 12

$$\boxed{\textbf{Do not calculate values. Make difference readable.}}$$

Conclusion

The executable sign

Tractatus notation begins with:

|

Python begins with signs, names, parentheses, functions, and lines.

Yet these forms too receive their meaning only through differences.

A name distinguishes a place.

A parenthesis binds an order.

A function marks a transition.

A condition opens or closes a gate.

A loop clocks a path.

A list carries a trace.

A return makes a reading public.

Thus Python does not become the origin of the difference.

It becomes a medium of its representation.

The complete connection is:

$$|\!\rightarrow \Delta \rightarrow I \rightarrow F_Q \rightarrow \gamma \rightarrow H_n \rightarrow \Omega \rightarrow Addr \rightarrow read$$

In Python:

```
relation = distinguish(left, right)

question = form_question(
    relation,
)

path = trace_path(
    question,
)

trace = integrate(
    path,
    through=gates,
)

address = locate(
    trace,
    relative_to=question,
)

label = read(
    address,
)
```

The code writes no finished object.

It holds the opening of the relation.

It orders the transitions.

It limits visibility.

It preserves the trace.

It reads the address.

It releases the label.

Thus the sign becomes an executable sign.

Thus the formula becomes an enactment.

Thus the difference becomes code without the difference becoming a thing in the code.

The formula shows the relation. **The code unfolds its reading.** **Both write the same difference.**