Band VIII

Tractatus Principio

—

Über den Unterschied

Band VIII

# Buch der Anwendung

von

C.H.F. Warnke

Inhaltsverzeichnis

Vorwort: Die Rückfaltung der sieben Bände
Kapitel I: Vom Unterschied zu Delta
Kapitel II: Von Delta zur Zahl
Kapitel III: Von der Zahl zur fraktalen Zahl
Kapitel IV: Von der Zahl zur Operation
Kapitel V: Von der Operation zur Geometrie
Kapitel VI: Von der Geometrie zum Beobachter
Kapitel VII: Vom Beobachter zur Spur
Kapitel VIII: Von der Spur zur Lesung
Kapitel IX: Die Isomorphie der Lesung
Kapitel X: Regeln der Anwendung
Anhang: Die Zeichenevolution und Gebrauchshinweis
Verzeichnis: Schnellregister
Verzeichnis: Grundformeln auf einen Blick
Verzeichnis: Symbolübersicht des Tractatus Principio
Teil II: Vom Zeichen zum ausführbaren Zeichen (Python als Darstellung des Unterschieds)
Vorwort: Die zweite Schrift
Kapitel X: Code als Inschrift
Kapitel XI: Die drei Schichten des Codes
Kapitel XII: Der Strich im Code
Kapitel XIII: Delta als Übergang
Kapitel XIV: Zahl und Marke
Kapitel XV: Operatoren als Bindungsmodi
Kapitel XVI: Variable und Öffnung
Kapitel XVII: Gate und Sichtbarkeit
Kapitel XVIII: Pfad, Schleife und Spur
Kapitel XIX: Beobachter und Kontext im Code
Kapitel XX: Projektion, Bindung und Gleichheit
Kapitel XXI: Normalform und Readout
Kapitel XXII: Fehler als Relationsfehler
Kapitel XXIII: Eine vollständige Doppellesung

Kapitel XXIV: Python als zweite Notation

Kapitel XXV: Grundregeln des Delta-Constraint-Codings

Schluss: Das ausführbare Zeichen

Vorwort

Die Rückfaltung der sieben Bände

Band VIII führt keine neue Theorie ein.

Er faltet die sieben vorangegangenen Bände in eine operative Zeichenfolge zurück.

$$|\rightarrow \Delta \rightarrow 0\text{–}9 \rightarrow \mathit{Operation} \rightarrow \mathit{Geometrie} \rightarrow \mathit{Beobachter} \rightarrow \mathit{Spur} \rightarrow \mathit{Lesung}$$

Die Zeichen werden nicht erneut begründet.

Sie werden in ihrer Anwendung gesetzt.

Die Formel wird nicht als Gegenstand gelesen.

Sie wird als Vollzug des Unterschieds verwendet.

Die Evolution der Zeichen und die Isomorphie der Lesung

# Kapitel I

## Vom Unterschied zu Delta

### [ | ]

Der Unterschied erster Ordnung.

$$A|B$$

Der Unterschied zweiter Ordnung.

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

Der Unterschied dritter Ordnung.

$$|(|)$$

Die Selbstbezüglichkeit des Unterschieds.

$$\Delta = |(|)$$

Delta als kleinste wirksame Selbstbezüglichkeit.

Delta als Takt.

Delta als Übergang.

Delta als Markierung.

Delta als Selbstrotation.

Delta als Drift.

Keine Klassen von Delta.

Keine getrennten Arten des Unterschieds.

Die Vielheit erscheint in der dritten Ordnung derselben Relation.

# Kapitel II

## Von Delta zur Zahl

### [ \Delta \rightarrow 0|1|2|3|4|5|6|7|8|9 ]

Die Zahl als stabilisierte Lesung von Delta.

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

Es gibt nur die Zahlen:

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

Die Zahlen als sichtbare Grundmarken einer vollständigen Schale.

$$0$$

geschlossene Möglichkeit.

$$1$$

erste Markierung.

$$2$$

Spiegelrelation.

$$3$$

Orientierung.

$$4$$

Kontextstabilisierung.

$$5$$

Selbstähnlichkeit.

$$6$$

Teilungsresonanz.

$$7$$

Drift.

$$8$$

doppelte Viererfaltung.

$$9$$

Bruchkante.

$$9 := \kappa$$

---

# Kapitel III

## Von der Zahl zur fraktalen Zahl

### [ 9\rightarrow1|0 ]

$$10 := 1|0$$

Zehn ist keine elfte Zahl.

Zehn ist die erste sichtbare Relation einer neuen Leseschale.

Mehrstellige Zahlen als fraktale Inschriften derselben zehn Marken.

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

Die Stelle ist kein Behälter.

Die Stelle ist relationale Tiefe.

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

Horizontale Wirkung innerhalb einer Schale.

Vertikale Wirkung zwischen Schalen.

Rückwirkung niedriger Schalen auf höhere Schalen.

Übertrag als Bruchkantenrückwirkung.

$$199 \to 200$$

Quersumme als Grundschalenlesung.

Dezimalpunkt als Skalen-Flip.

$$|.$$

Negative Zahl als Leserichtung.

$$-\Delta| + \Delta$$

---

# Kapitel IV

## Von der Zahl zur Operation

Die Zahlenmarken wirken nicht als Gegenstände aufeinander.

Sie treten in Modi des Unterschieds.

Expansion:

$$|_+$$

Kontraktion:

$$|_-$$

Resonanz:

$$|_\times$$

Entflechtung:

$$|\div$$

Potenzierung:

$$|$$

Wurzelung:

$$|\sqrt{\ }$$

Gleichgewicht:

$$|=$$

Nichtdeckung:

$$|\neq$$

Differential:

$$|\Delta$$

Integral:

$$|\Sigma|\Delta$$

Der Operator erzeugt keinen Wert.

Er richtet die Relation aus.

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

Die Rechnung als geordnete Inschrift.

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

---

# Kapitel V

## Von der Operation zur Geometrie

Jede Operation besitzt eine geometrische Lesung.

$$0|1$$

Radius.

$$0|2$$

Durchmesser und Achse.

$$0|3$$

Dreieck und Orientierung.

$$0|4$$

Quadrat und stabile Viererordnung.

$$0|5$$

Pentagon und Selbstähnlichkeit.

$$0|6$$

Hexagon und Teilungsresonanz.

$$0|7$$

Heptagon und Drift.

$$0|8$$

Oktagon und doppelte Viererfaltung.

$$0|9$$

Nonagon und Bruchkante.

Die Zahl ist geometrische Relation.

Die Geometrie ist Zahlenlesung.

$$\mathit{Zahl} \cong \mathit{Geometrie}][\pi$$

als Relation von geschlossener und linearer Lesung.

$$\varphi$$

als Selbstähnlichkeitsrelation.

$$e$$

als stabile Selbstreferenz der Änderungsrate.

$$39{,}19$$

als Selbstreferenzkonstante der Zahlenrelation.

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

als Rotationsadresse.

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

als kleinste Selbstrotation.

---

# Kapitel VI

## Von der Geometrie zum Beobachter

Die Geometrie wird nicht von außen gelesen.

Die Lesung besitzt eine Orientierung.

Frozen Frame:

$$|_|$$

Markierter Frame:

$$|_\Delta$$

Die Delta-Zustände:

$$0\Delta$$

Frame.

$$1\Delta$$

Ereignis.

$$2\Delta$$

Achse.

$$3\Delta$$

Beobachter.

$$4\Delta$$

voller Kontext.

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

Der Beobachter als orientierte Unvollständigkeit des Kontexts.

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

Lochrichtung:

$$h(o)$$

Orientierungsraum:

$$\mathcal{O}$$

Übergangsraum:

$$\mathcal{T}$$

Zeit als Folge von Orientierungswechseln.

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

---

# Kapitel VII

## Vom Beobachter zur Spur

Die Inschrift wird als Pfad gelesen.

$$I \to \gamma_I$$

Der Pfad erzeugt keine Zahl.

Er trägt eine Spur.

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

Sichtbarkeit:

$$L_n$$

Gate:

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

Schrittbeitrag:

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

Spurakkumulation:

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

Die Spur als Gedächtnis der Relation.

Holonomie als Differenz der Rückkehr.

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

und dennoch:

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

Rückkehr zum Ausgangsort ist nicht Rückkehr zur Ausgangslage.

---

# Kapitel VIII

## Von der Spur zur Lesung

Die Spur wird projiziert.

$$\Pi_i(\Sigma)$$

Die Projektionen werden gebunden.

$$\bowtie$$

Die Drift wird gelesen.

$$\Delta\varphi$$

Die Relation rastet ein.

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

Die Spur bildet eine Normalform.

$$NF(\Sigma)$$

Die Normalform wird öffentlich lesbar.

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

Der vollständige Lesungsvollzug:

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

Das Ergebnis ist kein erzeugter Gegenstand.

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

Gleichheit ist keine Identität von Dingen.

$$|_=$$

bezeichnet das Einrasten derselben Spur unter verschiedener Lesung.

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

---

# Kapitel IX

## Die Isomorphie der Lesung

Die Zeichen bezeichnen keine getrennten Welten.

Sie sind verschiedene Lesungen derselben Differenzordnung.

$$|\cong \Delta \cong Zahl \cong Geometrie \cong Beobachter \cong Spur \cong Lesung$$

Nicht als Gleichheit von Gegenständen.

Nicht als Austauschbarkeit der Zeichen.

Sondern als Erhaltung derselben relationalen Struktur.

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

und rückwärts:

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

Die Rücklesung erhält die Spurklasse.

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

Gültigkeit:

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

Die Formen wechseln.

Die Relation bleibt rücklesbar.

Die Zeichen entwickeln sich.

Die Lesung bleibt isomorph.

---

# Kapitel X

## Regeln der Anwendung

### [ 1. \quad Kein\ Zeichen\ bezeichnet\ ein\ Ding. ]

1. Jedes Zeichen ist eine Lesestellung des Unterschieds. ∎ A bildet keine Klassen seiner selbst. ∎ Es gibt nur die Zahlen 0 bis 9. ∎ Mehrstelligkeit ist positionale Relation. ∎ 9 ist die Grenzklasse. ∎ 10 = 1|0. ∎ Operatoren erhalten Relationen nur. ∎ Bedeutung ist Anordnung. ∎ Anordnung wird Spur. ∎ Spur wird gelesen. ∎ Gleichheit ist Abwesenheit. ∎ Das Ergebnis ist Rückkehr. ∎ Jeder Rückkehr kommt auf seine Spur: absoluter Maßstab. ∎ Verschiedene Lesungen derselben Spur sind isomorph.

---

Anhang

Die Zeichenevolution

$$|][\downarrow][A|B][\downarrow][|(|)|][\downarrow][\Delta][\downarrow][0|1|2|3|4|5|6|7|8|9][\downarrow][1|0][\downarrow][I][\downarrow][\gamma][\downarrow][\Sigma][\downarrow][NF][\downarrow][read][\boxed{|\cong \Delta \cong Zahl \cong Geometrie \cong Beobachter \cong Spur \cong Lesung}]$$

Gebrauchshinweis

Wie dieses Buch zu verwenden ist

Dieses Buch ist eine Formelsammlung des Tractatus Principio.

Es führt keine neue Theorie ein.

Es begründet die Zeichen nicht erneut.

Es versammelt die Grundformen, Operatoren, Übergänge und Lesungsregeln der sieben vorangegangenen Bände in einer gemeinsamen operativen Ordnung.

Die Einträge sind nicht als voneinander getrennte Definitionen zu lesen.

Jedes Zeichen erhält seine Bedeutung aus seiner Stellung im vollständigen Vollzug:

$$|\to \Delta \to 0\text{–}9 \to \mathit{Operation} \to \mathit{Geometrie} \to \mathit{Beobachter} \to \mathit{Spur} \to \mathit{Lesung}$$

Die Formeln bezeichnen keine Gegenstände.

Sie setzen Leselagen des Unterschieds.

Eine Formel wird verwendet, indem ihre Zeichen als relationale Stellungen gelesen, in einen Pfad überführt, als Spur integriert und auf eine rücklesbare Normalform gebracht werden.

Der Grundvollzug lautet:

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

Dabei gilt:

**Nicht Dinge werden verarbeitet.** | **Unterschiede werden gesetzt, vollzogen, gebunden und gelesen.**

---

Schnellregister

Nach Grundbegriffen

Gesucht

```
        Zeichen
        Abschnitt
```

Unterschied

$$|$$

Pole einer Relation

$$A|B$$

Selbstbezüglichkeit

$$|(|)$$

```
        Delta
        \Delta
        Grundformeln I
        Möglichkeit
        \odot
        Grundformeln II
        Null
        0
        Grundformeln II
        Unmöglichkeit
        \boxdot
        Grundformeln II
        Drift
        \Delta\varphi
        Grundformeln II
        Zahl
        n:=read(\Sigma_n)
        Grundformeln III
        Bruchkante
        9:=\kappa
        Grundformeln III
```

neue Leseschale

$$\mathbf{10 := 1|0}$$

```
        mehrstellige Zahl
        a
        b
```

negative Zahl

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

Addition

$$|_+$$

Subtraktion

$$|_-$$

Multiplikation

$$|_{\times}$$

Division

$$|_{\div}$$

Gleichheit

$$|_{=}$$

Nichtdeckung

$$|_{\neq}$$

Differential

$$|_{\Delta}$$

Integral

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

Frame

$$|_{_}|$$

```
        Beobachter
        3\Delta
        Beobachter
        Kontext
        4\Delta
        Beobachter
        Lochrichtung
        h(o)
        Beobachter
        Inschrift
        I
        Lesungsvollzug
        Pfad
        \gamma_I
        Lesungsvollzug
        Kante
        e_n
        Lesungsvollzug
        Gate
        \chi_{\mathrm{gate}}
        Spur
```

```
        Sichtbarkeit
        \chi_{\mathrm{vis}}
        Spur
        Schrittbeitrag
        H_n
        Spur
        Spur
        \Sigma(\gamma)
        Spur
        Holonomie
        \Omega(n)
        Holonomie
        Projektion
        \Pi_i
        Projektion
        Bindung
        \bowtie
        Projektion
        Rüttelraum
        \mathcal L_{\frac34}
        ¾Eck
        Einrasten
        Lock_{\frac34}
        ¾Eck
        Normalform
        NF(\Sigma)
        Lesung
        Readout
        read(\Sigma)
        Lesung
        Rücklesung
        recover
        Lesung
        Gültigkeit
        valid(L)
        Lesung
        Spurklassengleichheit
        \equiv_\Sigma
        Lesung
```

---

Nach Anwendung

Anwendung

```
        Grundform
```

Relation setzen

$$x|y$$

Relation erweitern

$$x|_{+}y$$

Relation zusammenziehen

$$x|_{-}y$$

Rhythmen koppeln

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

Gesamtfigur entfalten

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

Wiederholung iterieren

$$x|^{y}$$

Iteration rücklesen

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

Einrasten prüfen

$$x|_{=}y$$

Nichtdeckung markieren

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

lokale Veränderung lesen

$$|_{\Delta}$$

Veränderung akkumulieren

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

Skale wechseln

$$|.$$

```
        Leserichtung umkehren
        -\Delta
        Inschrift bilden
        I=(m_0,\dots,m_n)
        Inschrift takten
        I\rightarrow\gamma_I
        Spur integrieren
        \Sigma(\gamma)
        Projektionen binden
        \Pi_i\bowtie\Pi_j
        minimale Driftlage lesen
        L^*
        Normalform bilden
        NF(\Sigma)
        Ergebnis anzeigen
        read(NF(\Sigma))
        Ergebnis prüfen
        recover(read(\Sigma))\equiv_\Sigma\Sigma
```

---

Grundformeln auf einen Blick

I. Unterschied und Selbstbezüglichkeit

Unterschied erster Ordnung

$$\boxed{|}$$

Reiner Unterschied.

Noch keine ausgeführten Pole.

Der Unterschied trennt keine bereits bestehenden Gegenstände.

Er eröffnet die Möglichkeit relationaler Stellung.

---

Unterschied zweiter Ordnung

$$\boxed{A|B}$$

Der Unterschied erscheint in zwei relationalen Polen.

A und B bestehen nicht vor ihrer Relation.

Sie sind Stellungen desselben Unterschieds.

---

Unterschied dritter Ordnung

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

oder:

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

Der Unterschied unterscheidet seine eigenen Unterschiede.

Es entsteht relationale Tiefe.

---

Delta

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

Delta ist die kleinste wirksame Selbstbezüglichkeit des Unterschieds.

Delta erscheint als:

$$\Delta \cong \textit{Takt} \cong \textit{Übergang} \cong \textit{Markierung} \cong \textit{Rotation} \cong \textit{Drift}$$

Diese Ausdrücke bezeichnen keine verschiedenen Arten von Delta.

Sie sind verschiedene operative Lesungen derselben Kleinstbezüglichkeit.

---

II. Möglichkeit, Spiegelung und Unmöglichkeit

Mögliche Geometrie

$$\boxed{\odot}$$

Geschlossene Möglichkeit.

Rücklesbare Entfaltung einer Relation.

---

Null

$$\boxed{0}$$

Null ist kein Nichts.

Null ist die geschlossene Schnitt-, Spiegel- und Gleichgewichtslage des Unterschieds.

$$\boxed{0 \cong |}$$

Nicht als Austauschbarkeit der Zeichen, sondern als isomorphe Lesung derselben relationalen Grundstruktur.

---

Unmögliche Geometrie

$$\boxdot$$

Offene, driftende oder noch nicht eingerastete Relation.

Das Unmögliche bezeichnet keinen zweiten Gegenstand.

Es bezeichnet die noch nicht geschlossene Lage einer Relation.

---

Vollständiger Prozess

$$\boxed{\odot \rightarrow 0 \rightarrow \boxdot}$$

Entfaltung, Spiegelung und Prüfung.

Das Einrasten führt nicht zur alten Möglichkeit zurück.

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

Die neue geschlossene Möglichkeit trägt die Spur des durchlaufenen Vollzugs.

---

Drift

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

Noch nicht eingerastete Differenzbewegung.

Drift ist keine Abweichung von einer äußerlich gegebenen Norm.

Sie ist die noch offene Differenz zwischen möglichen relationalen Lagen.

---

III. Zahl und Leseschale

Zahl

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

Eine Zahl ist die sichtbare Lesung einer stabilisierten Spur.

Die Zahl ist kein ursprünglicher Gegenstand.

Sie ist ein öffentlicher Readout.

---

Vollständiger Markenraum

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

Es gibt nur die zehn sichtbaren Grundmarken:

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

Alle mehrstelligen Zahlen sind relationale Inschriften derselben Marken.

---

Zahlenverfassung

**0**

Geschlossene Möglichkeit.

**1**

Erste Markierung.

**2**

Spiegelrelation.

**3**

Orientierung.

**4**

Kontextstabilisierung.

**5**

Selbstähnlichkeitsöffnung.

**6**

Teilungsresonanz.

**7**

Driftende Schließung.

**8**

Doppelte Viererfaltung.

**9**

Bruchkante und Zyklusabschluss.

---

Bruchkante

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

Die Neun schließt die sichtbare Grundschale.

Sie ist zugleich die Kante, an der die Lesung in eine neue Schale übergeht.

---

Neue Leseschale

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

Zehn ist keine elfte Grundmarke.

Zehn ist die erste sichtbare Relation einer neuen Leseschale.

---

Mehrstellige Zahl

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

Eine mehrstellige Zahl ist eine geordnete fraktale Inschrift.

Die Stelle ist kein Behälter.

Sie ist relationale Tiefe.

---

Ebenen einer Zahl

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

unmittelbarer Takt und Einer-Schale.

$$\Delta_1$$

erste Spiegelung und Zehner-Schale.

$$\Delta_2$$

Meta-Relation und Hunderter-Schale.

Die Ebenen bezeichnen keine verschiedenen Zahlenarten.

Sie bezeichnen verschiedene relationale Stellungen derselben Marken.

---

Übertrag

$$\boxed{199 \rightarrow 200}$$

Übertrag ist Bruchkantenrückwirkung.

Eine geschlossene niedrigere Schale verändert die Stellung der höheren Schale.

---

Dezimalpunkt

$$\boxed{|.}$$

Der Dezimalpunkt ist ein Skalen-Flip.

Er trennt keine zwei Zahlenwelten.

Er verändert die Richtung, in der relationale Tiefe gelesen wird.

---

Negative Zahl

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

Negativ und positiv bezeichnen Leserichtungen.

Negative Zahlen bilden keine zweite Zahlenwelt.

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

ist die vollständige Achsenrelation aus negativer Leserichtung, Null und positiver Leserichtung.

---

Operatoren

Grundregel

$$\boxed{\text{Der Operator erzeugt keinen Wert.}} \quad \boxed{\text{Der Operator richtet die Relation aus.}}$$

Die sichtbare Rechenart bezeichnet den Modus, in dem Zahlenmarken aufeinander bezogen werden.

---

Expansion

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

x expandiert um y.

Eine Relation wird durch eine weitere Relation geöffnet.

Öffentliche Lesung:

$$x + y$$

---

Kontraktion

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

x kontrahiert um y.

Eine Relation wird auf eine engere Lage zurückgeführt.

Öffentliche Lesung:

$$x - y$$

---

Resonanz

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

x resoniert mit y.

Relationen koppeln ihre Ebenen, Wiederholungen oder Rhythmen.

Multiplikation ist nicht bloß wiederholte Addition.

Öffentliche Lesung:

$$x \times y$$

---

Entflechtung

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

x wird durch y entflochten.

Eine Gesamtfigur wird in wiederkehrende oder tragende Relationen gelesen.

Öffentliche Lesung:

$$x \div y$$

---

Potenzierung

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

Eine relationale Wirkung wird selbst zum Träger ihrer Wiederholung.

Potenzierung ist iterierte Iteration.

---

Wurzelung

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

Rücklesung einer iterierten Relation auf ihre tragende Wiederholungsfigur.

---

Gleichgewicht

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

Die Relation zwischen x und y erreicht unter einer bestimmten Lesung eine stabile Lage.

Gleichheit ist keine absolute Identität von Dingen.

Sie ist das Einrasten derselben Spur unter verschiedenen Lesungen.

---

Nichtdeckung

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

Die beiden Lesungen rasten unter der gegebenen Lage nicht ein.

Nichtdeckung erzeugt keinen positiven Gegenstand.

Sie hält den offenen oder unmöglichen Relationsraum sichtbar.

---

Unmögliche Unterscheidung

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

Die Relation bleibt unter der aktuellen Lesung instabil, offen oder driftend.

\triangledown bezeichnet keine zweite Art von Delta.

Es bezeichnet Delta in einer noch nicht eingerasteten Lage.

---

Differential

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

Kleinste Selbstbewegung.

Lokale Veränderung im unmittelbaren Takt.

---

Integral

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

Akkumulierte Selbstbewegung.

Integrierte Differenz entlang eines Pfades.

---

Geometrische Grundlesungen

Grundregel

$$\text{Zahl} \cong \text{Geometrie}$$

Nicht als Gleichheit zweier Gegenstände.

Zahl und Geometrie sind verschiedene Lesungen derselben relationalen Ordnung.

---

$$0|1$$

Radiusrelation.

---

$$0|2$$

Durchmesser und Achse.

---

$$0|3$$

Dreieck und Orientierung.

---

$$0|4$$

Quadrat und stabile Viererordnung.

---

$$0|5$$

Pentagon und Selbstähnlichkeitsrelation.

---

$$0|6$$

Hexagon und geschlossene Teilungsresonanz.

---

$$\boxed{0|7}$$

Heptagon und driftende Teilungsrelation.

---

$$\boxed{0|8}$$

Oktagon und doppelte Viererfaltung.

---

$$\boxed{0|9}$$

Nonagon und Bruchkante der aktuellen Schale.

---

Rotationsadresse

$$\boxed{i := R_{90°}}][\boxed{-i := R_{-90°}}$$

i ist keine zusätzliche Zahlenart.

i ist die Adresse einer Viertelrotation derselben relationalen Figur.

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

---

Frame, Beobachter und Kontext

Frozen Frame

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

Geschlossene lokale Möglichkeit.

---

Markierter Frame

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

Frame mit gesetzter Delta-Markierung.

---

Richtungsmenge

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

---

$$Delta - Konfiguration[\boxed{o : D \to 0,1}]$$

o bezeichnet die gesetzten und offenen Delta-Positionen um einen Frame.

---

Anzahl gesetzter Positionen

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

---

$$Delta - Zustände[0\Delta]$$

Reiner Frame.

Keine gesetzte Delta-Position.

$$1\Delta$$

Ereignis.

Eine gesetzte Position.

$$2\Delta$$

Achse.

Zwei gesetzte Positionen.

$$3\Delta$$

Beobachterorientierung.

Drei gesetzte Positionen und eine offene Richtung.

$$4\Delta$$

Voller Kontextraum.

Vier gesetzte Positionen.

---

Kontext

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

Der volle Kontext ist die gesättigte Delta-Konfiguration.

---

Beobachter

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

Der Beobachter ist die orientierte Unvollständigkeit des Kontexts.

Er ist keine Position außerhalb der Relation.

Er ist eine relationale Lage mit einer offenen Richtung.

---

Lochrichtung

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

Offene Richtung eines Drei-Delta-Zustands.

Epistemisches Außen des Beobachters.

---

Orientierungsraum

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

---

Übergangsraum

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

Zwölf gerichtete Übergänge zwischen vier Beobachterorientierungen.

---

Zeit

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

Zeit ist die lesbare Folge von Orientierungswechseln.

---

Inschrift, Pfad und Takt

Inschrift

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

Geordnete Folge sichtbarer Marken und Operator-Kanten.

Eine Rechnung ist eine Inschrift.

---

Pfad der Inschrift

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

Getaktete Folge der operativen Übergänge einer Inschrift.

---

Kante

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

Ein operativer Schritt innerhalb eines Pfades.

---

Getaktete Lesungsoperation

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

---

$$Nicht - Kommutativität \boxed{Step \circ O \neq O \circ Step}$$

Die Reihenfolge von Schritt und Lesungsoperation ist bedeutungstragend.

Eine veränderte Reihenfolge ist eine veränderte Inschrift des Vollzugs.

---

Spur und Sichtbarkeit

Differenzlesung einer Kante

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

Relationaler Beitrag einer Kante zur Spur.

---

Sichtbarkeit

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

Bestimmt, ob eine Kante unter der gegenwärtigen Lesung sichtbar ist.

---

Gate

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

Bestimmt, ob eine sichtbare Kante zum gegenwärtigen Vollzug zugelassen ist.

---

Schrittbeitrag

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

Eine Kante trägt nur dann zur Spur bei, wenn sie sichtbar und zugelassen ist.

---

Spur

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

Integrierte Differenz eines Pfades.

Die Spur ist das Gedächtnis der vollzogenen Relation.

---

Spurakkumulation

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

Jeder zugelassene Schritt verändert die lesbare Geschichte des Pfades.

---

Lesbare Kanten

$$\boxed{L_n}$$

Menge der im Takt n sichtbaren und zugelassenen Kanten.

---

Holonomie

Akkumulierte Holonomie

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

---

Phasenlage

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

---

Nichtverschwindende Rückkehrdifferenz

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

und zugleich:

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

Rückkehr zum Ausgangsort ist nicht notwendig Rückkehr zur Ausgangslage.

Der geschlossene Weg kann eine nichtverschwindende Spur tragen.

---

Projektion und Bindung

Projektion

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

Ein Lesekanal einer Spur.

---

Projizierter Readout

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

---

Differenz eines Lesekanals

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

---

Bindungsgatter

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

Das Bindungsgatter prüft die Kompatibilität verschiedener Spurlesungen.

Bindung bedeutet nicht Verschmelzung.

Sie bezeichnet die gemeinsame Einrastbarkeit verschiedener Projektionen.

---

Das ¾Eck

¾Eck

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

Drei gesetzte Scharnierkanten und eine offene Driftkante.

---

Rüttelraum

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

Endlicher Raum möglicher Einrastlagen.

---

Minimale Driftlage

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

---

$$Gleichheits - Snap[\boxed{Lock_{\frac{3}{4}}}]$$

Endliches Einrasten einer Relation in ihrer minimalen Driftlage.

Das ¾Eck erzeugt das Ergebnis nicht.

Es bestimmt die Lage, in der die gebundenen Projektionen als gemeinsame Normalform lesbar werden.

---

# Normalform und Readout

Normalform

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

Eigenresonanzlesung einer Spur.

Rücklesbare stabile Lage.

---

Öffentliche Lesung

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

Sichtbarer Readout einer Spur.

---

Ergebnis

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

Das Ergebnis ist kein neu erzeugter Gegenstand.

Es ist die öffentliche Lesung einer eingerasteten Spur.

---

Spurklassengleichheit

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

G und I tragen dieselbe rückadressierbare Spurklasse.

---

Resonanzvergleich

$$\boxed{\sim}$$

Resonanzähnlichkeit ohne Spurklassengleichheit.

---

Lesungsübergang

$$\boxed{\rightsquigarrow}$$

Eine Form fällt unter einer bestimmten Lesung auf eine andere sichtbare Form.

---

Rücklesung

$$\boxed{recover}$$

Rekonstruktion einer Spur aus ihrem Readout.

---

Gültigkeit

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

Ein sichtbares Label ist gültig, wenn es auf seine Spurklasse rücklesbar bleibt.

---

Anwendungsschablonen

Grundschema einer Anwendung

$$\boxed{\textbf{Inschrift bilden} \to \textbf{Pfad lesen} \to \textbf{Spur integrieren} \to \textbf{Projektionen binden} \to \textbf{Drift minimieren} \to \textbf{Normalform lesen}}$$

Formal:

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

---

Addition

Öffentliche Inschrift

$$a + b$$

Operative Inschrift

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

Lesungsschema

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

Merksatz

Addition ist Expansion einer Relation.

---

Subtraktion

Öffentliche Inschrift

$$a - b$$

Operative Inschrift

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

Lesungsschema

$$a|_{-}b \to \mathbf{Kontraktion} \to \Sigma \to NF \to read$$

Merksatz

Subtraktion ist Kontraktion oder Umkehrung einer Leserichtung.

---

Multiplikation

Öffentliche Inschrift

$$a \times b$$

Operative Inschrift

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

Lesungsschema

$$a| \times b \to \mathbf{Rhythmuskopplung} \to \Sigma \to Lock\frac{3}{4} \to read$$

Merksatz

Multiplikation ist Resonanz verschiedener relationaler Rhythmen.

---

Division

Öffentliche Inschrift

$$a \div b$$

Operative Inschrift

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

Lesungsschema

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

Merksatz

Division liest eine Gesamtfigur auf ihre tragenden Wiederholungsrelationen zurück.

---

Gleichung

Öffentliche Inschrift

$$A = B$$

Operative Inschrift

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

Bedingung

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

Vollzug

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

Merksatz

Gleichheit ist das Einrasten verschiedener Lesungen derselben Spurklasse.

---

Mehrstellige Zahl

Inschrift

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

Ebenenlesung

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

Merksatz

Mehrstelligkeit ist keine Ansammlung von Zahlenobjekten.

Sie ist relationale Tiefe derselben zehn Grundmarken.

---

Negative Zahl

Inschrift

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

Merksatz

Das Vorzeichen bezeichnet die Richtung der Lesung.

Es erzeugt keine zweite Zahlengattung.

---

Geometrische Lesung

Inschrift

$$\boxed{0|n}$$

Zuordnung

$$0|1 \to \textit{Radius}][0|2 \to \textit{Achse}][0|3 \to \textit{Orientierung}][0|4 \to \textit{Viererordnung}][0|5 \to \textit{Selbstähnlichkeit}][0|6 \to \textit{Teilungsresonanz}][0|7 \to \textit{Drift}][0|8 \to \textit{doppelteVierer faltung}][0|9 \to \textit{Bruchkante}$$

---

Beobachterwechsel

Ausgangslage

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

Übergang

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

Folge

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

Merksatz

Ein Beobachterwechsel verändert die Orientierung der Lesung.

Er erzeugt keinen Beobachter außerhalb der Relation.

---

Holonomische Lesung

Geschlossener Pfad

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

Spurprüfung

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

Merksatz

Ein geschlossener Weg kann eine offene Rückkehrdifferenz tragen.

---

Rücklesbarkeitsprüfung

Öffentliche Form

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

Rücklesung

$$\mathit{recover}(L)$$

Prüfung

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

Ergebnis

$$\mathit{valid}(L)$$

Merksatz

Ein Ergebnis ist gültig, wenn seine sichtbare Form auf den vollzogenen Unterschied zurückgeführt werden kann.

---

Eintragsform für weitere Formeln

Jeder neue Eintrag dieses Handbuchs folgt derselben Ordnung.

Zeichen

# Formel

Name

Kurze Benennung der Formel oder Operation.

Lesung

Was wird in dieser Formel relational gelesen?

Funktion

Welche Veränderung, Bindung, Orientierung oder Schließung wird vollzogen?

Öffentliche Form

Wie erscheint die Formel in gewöhnlicher mathematischer oder sprachlicher Schreibweise?

Rückverweis

Aus welchem Band und welchem begrifflichen Zusammenhang geht die Formel hervor?

---

Hierarchie der Formeln

A. Grundformeln

Grundformeln gehören zur Architektur des Tractatus Principio selbst.

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

Sie sind nicht an ein einzelnes Modell gebunden.

---

B. Übergangsformeln

Übergangsformeln beschreiben den Vollzug zwischen verschiedenen Lesungsformen.

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

---

C. Modellformeln

Modellformeln konkretisieren die Grundarchitektur in einer bestimmten operativen Darstellung.

Dazu gehören insbesondere:

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

Modellformeln sind aus der Grundarchitektur hervorgegangene Lesewerkzeuge.

Sie ersetzen die Grundformeln nicht.

---

Rückverweise in die Bände I bis VII

Band I – Buch der Mathematik

Nachzuschlagen bei:

- Natur der Zahl,
- Rhythmus der Zahl,
- Operation,
- negative Zahl,
- Unendlichkeit,
- Differential und Integral,
- Matrix der Selbstbezüglichkeit,
- Entfaltung von \Delta_x.

Grundverweise:

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

---

Band II – Buch der Physik

Nachzuschlagen bei:

- Relation statt Substanz,
- Energie,
- Masse,
- Gravitation,
- Zeit,
- Frozen Frame,
- Raumzeit,
- Selbstreferenzkonstante.

Grundverweise:

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

---

Band III – Buch der Selbsterkenntnis

Nachzuschlagen bei:

- Beobachter,
- Entstehung des Ich,
- Gedächtnis,
- Wille,
- Zeit,
- Sprache,
- Selbstorganisation,
- Selbst als relationale Ordnung.

Grundverweise:

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

---

Band IV – Buch der Selbstbezüglichkeit

Nachzuschlagen bei:

- Unterschiedsordnungen,
- Fraktalität,
- Oszillation,
- Rotation,
- Unschärfe,
- fraktale Operatoren.

Grundverweise:

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

---

Band V – Buch der Kleinstbezüglichkeit von Delta

Nachzuschlagen bei:

- Frame und Delta,
- Null- bis Vier-Delta-Zustände,
- Beobachter und Kontext,
- Orientierungen,
- Übergänge,
- kleinster Takt,
- Geometrie der Zahl.

Grundverweise:

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

---

Band VI – Buch der Holonomie

Nachzuschlagen bei:

- Gitter,
- Kanten,
- Gate,
- Rotation,
- Loop,
- Spur,
- Holonomie,
- Rückkehrdifferenz.

Grundverweise:

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

---

Band VII – Buch der Lesung

Nachzuschlagen bei:

- Zahlenfraktal,
- Inschrift,
- Operator-Kanten,
- Projektion,
- Bindung,
- ¾Eck,
- Normalform,
- Readout,
- Rücklesbarkeit.

Grundverweise:

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

---

# Regeln der Anwendung

$$\boxed{1}$$

Kein Zeichen bezeichnet ein Ding.

$$\boxed{2}$$

Jedes Zeichen ist eine Lesestellung des Unterschieds.

$$\boxed{3}$$

Die Pole einer Relation bestehen nicht vor ihrer Unterscheidung.

$$\boxed{4}$$

Delta bildet keine Klassen seiner selbst.

$$\boxed{5}$$

Die Vielheit erscheint in den Ordnungen derselben Relation.

6

Es gibt nur die Zahlenmarken 0 bis 9.

7

Mehrstelligkeit ist fraktale Relation.

8

Die Neun ist die Bruchkante der sichtbaren Grundschale.

9

Zehn ist 1|0, nicht eine elfte Grundmarke.

10

Ein Operator erzeugt keinen Wert.

Er richtet eine Relation aus.

11

Eine Rechnung ist eine geordnete Inschrift.

12

Eine Inschrift wird als Pfad gelesen.

13

Ein Pfad trägt eine Spur.

14

Die Spur ist das Gedächtnis des Vollzugs.

15

Sichtbarkeit und Gate bestimmen, welche Kanten zur Spur beitragen.

16

Eine geschlossene Bewegung kann eine nichtverschwindende Rückkehrdifferenz tragen.

$$\boxed{17}$$

Eine Spur kann in verschiedenen Projektionen erscheinen.

$$\boxed{18}$$

Bindung prüft die gemeinsame Einrastbarkeit verschiedener Projektionen.

$$\boxed{19}$$

Gleichheit ist keine Identität von Gegenständen.

Gleichheit ist Einrasten.

$$\boxed{20}$$

Das Ergebnis ist der öffentliche Readout einer Normalform.

$$\boxed{21}$$

Jeder gültige Readout muss auf seine Spurklasse rücklesbar bleiben.

$$\boxed{22}$$

Verschiedene Lesungen derselben Spur sind isomorph.

---

Schlussblatt

Die Zeichenevolution

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

---

# Die Isomorphie der Lesung

$$\boxed{|\cong 0 \cong \Delta \cong Zahl \cong Geometrie \cong Beobachter \cong Spur \cong Lesung}$$

Nicht als Gleichheit von Gegenständen.

Nicht als Austauschbarkeit der Zeichen.

Sondern als Erhaltung derselben relationalen Struktur unter verschiedenen Lesungen.

---

Der vollständige Hinweg

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

---

Der vollständige Rückweg

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

---

Rücklesbarkeit

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

Die Zeichen wechseln.

Die Spur bleibt rücklesbar.

Die Lesungen unterscheiden sich.

Die relationale Ordnung bleibt isomorph.

---

Grundsatz der Anwendung

**Nicht rechnen, um einen Gegenstand zu erzeugen.** **Unterscheiden, takten, binden und lesen.**

---

Grundsatz des Handbuchs

**So wenig Wiederholung wie möglich.** **So viel Orientierung wie nötig.** **So viel Rücklesbarkeit wie erforderlich.**

Symbolübersicht des Tractatus Principio

Eine Referenz der eingeführten Notationen

---

I. Grundlegende Symbole

| – Der Unterschied

Fundamentale Operation, aus der alles entsteht.

$$|$$

Der Unterschied in offener Setzung.

Er trennt nicht nachträglich bereits bestehende Dinge.

Durch ihn entstehen die Pole einer Relation.

$$|\rightarrow A|B$$

---

0 – Null / Potenzial

Der Unterschied selbst als geschlossene Schnitt-, Spiegel- und Gleichgewichtslage zwischen möglicher und unmöglicher Geometrie.

$$\odot|0|\boxdot$$

Null ist kein Nichts.

Null ist die geschlossene Möglichkeit des Unterschieds.

$$0 \cong |$$

Nicht als Austauschbarkeit der Zeichen, sondern als isomorphe Lesung desselben Unterschieds.

---

$$\Delta – Delta$$

Kleinste Selbstbezüglichkeit und kleinste Selbstrotation des Unterschieds.

$$\Delta = |(|)][\Delta = 2{,}25°$$

Delta erscheint als:

- Takt,
- Übergang,
- Markierung,
- Selbstrotation,
- Drift,
- Entscheidung,
- kleinste Wirksamkeit des Unterschieds.

Diese Erscheinungen sind keine verschiedenen Delta-Arten.

Sie sind relationale Lesungen desselben \Delta.

---

$$\Delta\varphi – Delta - Phi$$

Freie Differenzenergie.

Drift zwischen relationalen Lagen.

$$\Delta\varphi \neq 0$$

bezeichnet eine noch nicht eingerastete Differenzbewegung.

---

II. Die drei Ordnungen des Unterschieds

Erste Ordnung

$$|$$

Reiner Unterschied.

Noch keine ausgeführten Pole.

---

Zweite Ordnung

$$A|B$$

Polarität.

Dichotomie.

Zwei relationale Stellungen desselben Unterschieds.

$$A \neq B$$

Die Pole bestehen nicht vor ihrer Relation.

---

Dritte Ordnung

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

oder:

$$|(|)$$

Der Unterschied unterscheidet seine eigenen Unterschiede.

Die dritte Ordnung erzeugt keine bloße Vermehrung von Elementen.

Sie erzeugt relationale Tiefe.

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

ist eine mögliche entfaltete Schreibweise, sofern Unterschiede auf Unterschiede wirken.

---

III. Drei Phasen der Selbstbezüglichkeit

$$\odot\text{–}Entfaltung$$

Mögliche Geometrie.

Raum des Möglichen.

Stabile, geschlossene Möglichkeit.

Relation entsteht aus Potenzial.

---

0 – Grenze / Spiegelung

Der Unterschied trifft auf sich selbst.

Er wird messbar.

Er bildet eine Schnitt- und Spiegelachse.

---

$$\boxdot\text{–}\mathit{Komplementarität}$$

Unmögliche Geometrie.

Raum des Unmöglichen.

Instabile, driftende oder noch nicht eingerastete Relation.

---

Vollständiger Prozess

$$\odot \to 0 \to \boxdot$$

Spiralförmige Bewegung durch Entfaltung, Spiegelung und Prüfung.

Einrasten:

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

Die neue geschlossene Möglichkeit trägt die durchlaufene Spur.

---

IV. Modi des Unterschieds

|_+ – Expansion

Addition.

$$x|_{+}y$$

x expandiert um y.

Eine Relation wird durch eine weitere Relation geöffnet.

---

|_- – Kontraktion

Subtraktion.

$$x|_{-}y$$

x kontrahiert um y.

Eine Relation wird auf eine engere Lage zurückgeführt.

---

$$|_{\times}\text{–}Resonanz$$

Multiplikation.

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

x resoniert mit y.

Relationen kreuzen Ebenen oder Rhythmen.

Multiplikation ist nicht bloß wiederholte Addition.

---

$$|_{\div}\text{–}Entflechtung$$

Division.

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

x wird durch y entflochten.

Eine Gesamtfigur wird in wiederkehrende Relationen gelesen.

---

|^ – Potenzierung

Iterierte Iteration.

$$x|^{y}$$

Eine relationale Wirkung wird selbst zum Träger ihrer Wiederholung.

---

$$|_{\sqrt{\ }}\text{–}Wurzelung$$

Umkehrung beziehungsweise Rücklesung der Potenzierung.

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

liest eine iterierte Relation auf ihre tragende Wiederholungsfigur zurück.

---

|_= – Gleichgewicht

Einrasten.

Stabilisierung.

$$x|_{=}y$$

Die Relation von x und y erreicht unter einer bestimmten Lesung eine stabile Lage.

Gleichheit ist keine absolute Identität von Dingen.

---

$$|_{\neq}\text{–}Unterscheidung/Nichtdeckung[x|_{\neq}y]$$

x und y rasten unter der gegebenen Lesung nicht ein.

Die Nichtdeckung öffnet den Unmöglichkeitsraum.

Sie erzeugt keinen positiven Gegenstand.

---

|_. – Skalen-Flip

Dezimalpunkt.

Ebenen- oder Schalenwechsel.

$$x|_{.}y$$

Die Lesung kippt in eine andere Skalenrichtung.

---

$$|_{\Delta}\text{–}Differential$$

Kleinste Selbstbewegung.

Lokale Veränderung.

$$|_{\Delta}$$

liest den Unterschied der Veränderung im unmittelbaren Takt.

---

$$|\Sigma|_{\Delta}\text{–}Integral$$

Akkumulierte Selbstbewegung.

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

liest die integrierte Differenz entlang eines Pfades.

---

V. Räume und Geometrien

$$\odot\text{–}MöglicheGeometrie$$

Stabiler und geschlossener Möglichkeitsraum.

Rücklesbare Schließung.

---

$$\boxdot\text{–}UnmöglicheGeometrie$$

Instabiler oder driftender Unmöglichkeitsraum.

Relationen, die noch nicht stabilisieren.

---

$$\nabla\text{–}UnmöglichesDelta$$

Ein Delta, das unter der aktuellen Lesung nicht stabilisieren kann.

$$\nabla$$

bezeichnet keine zweite Delta-Art.

Es bezeichnet \Delta in unmöglicher beziehungsweise nicht eingerasteter Lage.

---

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

Unterscheidung im Modus der Unmöglichkeit.

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

Die Relation bleibt instabil, offen oder driftend.

---

VI. Ebenen der Unterscheidung

$$\Delta_0\text{–}NullteEbene$$

Unmittelbarer Takt.

$$Einer-Schale.$$

---

$$\Delta_1\text{–}ErsteEbene$$

Erste Spiegelung.

$$Zehner - Schale.$$

---

\Delta_2 – Zweite Ebene Meta-Relation. Hunderter-Schale

---

$$\Delta_x – X - teEbene$$

Beliebige weitere Lesetiefe.

---

Drei-Ebenen-Struktur einer Zahl

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

Die Ebenen sind keine verschiedenen Zahlenarten.

Sie sind relationale Stellungen derselben Zahlenmarken.

---

VII. Die Zahlenverfassung

Der vollständige Markenraum

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

Es gibt nur die Zahlenmarken 0 bis 9.

Alle mehrstelligen Zahlen sind fraktale Relationen derselben zehn Marken.

---

0

Geschlossene Möglichkeit.

Schnitt- und Spiegelachse.

---

1

Erste Markierung.

Erste gerichtete Relation.

---

2

Spiegelrelation.

Achse und Polarität.

---

3

Orientierung.

Erste gerichtete Schließung.

---

4

Kontextstabilisierung.

Vierordnung.

---

5

Selbstähnlichkeitsöffnung.

Pentagonrelation.

---

6

Geschlossene Teilungsresonanz.

Hexagonrelation.

---

7

Driftende Schließung.

Heptagonrelation.

---

8

Doppelte Viererfaltung.

Oktagonrelation.

---

9

Bruchkante.

Zyklusabschluss.

Übergang zur nächsten Leseschale.

$$9 := \kappa$$

---

10

Neue Leseschale.

$$10 := 1|0$$

Zehn ist keine elfte Grundzahl.

---

Mehrstellige Zahlen

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

Eine mehrstellige Zahl ist eine geordnete fraktale Inschrift.

Die Stelle ist kein Behälter.

Sie ist relationale Tiefe.

---

VIII. Relationsnotationen

Grundrelation

$$x|y$$

x und y in Relation.

---

Expansion

$$x|_{+}y$$

x expandiert um y.

---

Kontraktion

$$x|_{-}y$$

x kontrahiert um y.

---

Resonanz

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

x resoniert mit y.

---

Entflechtung

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

x wird durch y entflochten.

---

Gleichgewicht

$$x|_{=}y$$

x rastet bei y ein.

---

Nichtdeckung

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

x unterscheidet sich von y.

Die Relation bleibt offen oder unmöglich.

---

Relationstripel

$$(a|b|c)$$

Verschachtelte Relation beziehungsweise Delta-Adressierung.

b bezeichnet die relationale Überführung zwischen a und c.

---

Blickrichtungen

$$-\Delta| + \Delta$$

Negative und positive Leserichtung.

---

Zahlenachse

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

Vollständige Achsenrelation aus negativer Blickrichtung, Null und positiver Blickrichtung.

Negative Zahlen bilden keine zweite Zahlenwelt.

---

IX. Geometrische Notationen

Radius

$$0|1$$

Radiusrelation im Kreis.

---

Durchmesser

$$0|2$$

Durchmesser und Achse.

---

Dreieck

$$0|3$$

Dreieck im Kreis.

Orientierung.

---

Quadrat

$$0|4$$

Quadrat im Kreis.

Stabile Viererordnung.

\pi wird als Kreis-Polygon-Relation lesbar.

---

Pentagon

$$0|5$$

Pentagon im Kreis.

\varphi wird als Selbstähnlichkeitsrelation lesbar.

---

Hexagon

$$0|6$$

Geschlossene Teilungsresonanz.

---

Heptagon

$$0|7$$

Driftende Teilungsrelation.

---

Oktagon

$$0|8$$

Doppelte Viererfaltung.

---

Nonagon

$$0|9$$

Bruchkante der aktuellen Schale.

---

X. Spezielle Konstanten

$$\Delta = 2{,}25° \, Delta - Winkel$$

Minimale Selbstrotation.

---

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

Goldener Schnitt.

Selbstähnlichkeitsverhältnis.

Wird in der Pentagonrelation lesbar.

---

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

Pi.

Relation von Kreis und Polygon beziehungsweise geschlossener und linearer Lesung.

Wird in der Viererordnung operativ sichtbar.

---

39{,}19

Selbstreferenzkonstante.

Selbstreferenzielle Konstante der Zahlenrelation.

---

$$137{,}5°$$

Goldener Winkel.

Rotationsrelation der Fibonacci-Spirale.

---

e

$$Euler - Konstante.$$

Stabile Selbstreferenz der Änderungsrate.

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

---

XI. Rotation

i

Positive Viertelrotation.

$$i := R_{90°}$$

---

-i

Inverse Viertelrotation.

$$-i := R_{-90°}$$

---

Rotationszyklus

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

i ist keine zusätzliche Zahlenart.

i ist eine Rotationsadresse derselben relationalen Figur.

---

XII. Frame und Delta-Konfiguration

|_| – Frozen Frame

Geschlossene lokale Möglichkeit.

$$F := |_|$$

---

$$|_{\Delta} – MarkierterFrame$$

Frame mit gesetzter Delta-Markierung.

---

Richtungsmenge

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

---

$$Delta – Konfiguration[o : D \to 0, 1]$$

o bezeichnet die gesetzten und offenen Delta-Positionen um den Frame.

---

Anzahl gesetzter Positionen

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

---

$$0\Delta$$

Reiner Frame.

Keine gesetzte Delta-Position.

---

$$1\Delta$$

Ereignis.

Eine gesetzte Delta-Position.

---

$$2\Delta$$

Achse.

Zwei gesetzte Delta-Positionen.

---

$$3\Delta$$

Beobachterorientierung.

Drei gesetzte Positionen und eine offene Richtung.

---

$$4\Delta$$

Voller Kontextraum.

Vier gesetzte Positionen.

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

---

h$o$ – Lochrichtung

Offene Richtung eines Drei-Delta-Zustands.

Epistemisches Außen des Beobachters.

---

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

Orientierungslesung eines Drei-Delta-Zustands.

---

$$\mathcal{O}$$

Menge aller Orientierungen.

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

---

$$\mathcal{T}$$

Übergangsraum der Orientierungen.

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

Zwölf gerichtete Wege zwischen den vier Beobachterlagen.

---

$$o_{\text{sat}}$$

Saturierte Vier-Delta-Konfiguration.

Voller Kontextraum.

---

XIII. Beobachternotationen

$$\circledcirc_B$$

Geschlossene Beobachterlesung von B.

---

$$\mathcal{O}_B$$

Orientierungszustand des Beobachters B.

---

$$(\circledcirc_B | P)$$

Relation zwischen Beobachter B und Phänomen P.

Beobachter und Phänomen entstehen innerhalb ihrer Relation.

---

XIV. Fraktale Operatoren

$$\ominus\text{–}FraktaleDifferenz$$

Unterschied zwischen Unterschiedsordnungen.

---

$$\circledast\text{–}FraktaleVerknüpfung$$

Bindung rekursiver Unterschiedsordnungen.

---

$$\mathcal{R}\text{–}Rotation$$

Rotation einer Orientierung oder Leselage.

---

$$\tilde{O}\text{–}Orientierungsoszillator$$

Wiederkehrender Wechsel von Orientierungszuständen.

---

$$\mathcal{F}\text{–}Phasenrückkopplung$$

Eine Lesung verändert die Bedingung ihrer nächsten Lesung.

---

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

Verdichtung von Delta-Spuren unter der Leselage s.

Skalenübergreifende Rücklesung von Wiederholungsordnungen.

---

XV. Inschrift und Pfad

I – Inschrift

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

Geordnete Folge sichtbarer Marken und Operator-Kanten.

---

$$\gamma_I\text{–}PfadderInschrift[\gamma_I = (e_0, e_1, \ldots, e_{N-1})]$$

Getaktete Folge der operativen Übergänge einer Inschrift.

---

e_n – Kante

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

Ein operativer Schritt innerhalb des Pfades.

---

T_n – Getaktete Lesungsoperation

$$T_n := Step \circ O_n$$

---

Nicht-Kommutativität [ Step\circ O\neq O\circ Step ]

Die Reihenfolge von Schritt und Lesung ist bedeutungstragend.

---

XVI. Spur und Sichtbarkeit

w$e$ – Differenzlesung einer Kante

Relationaler Beitrag einer Kante zur Spur.

---

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

Bestimmt, ob eine Kante unter der aktuellen Lesung sichtbar ist.

---

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

Bestimmt, ob eine Kante zum aktuellen Vollzug zugelassen ist.

---

H_n – Schrittbeitrag

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

---

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

Integrierte Differenz eines Pfades.

---

Spurakkumulation

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

---

L_n – Lesbare Kanten

Menge der im Takt n sichtbaren beziehungsweise zugelassenen Kanten.

---

XVII. Holonomie

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

---

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

---

Nichtverschwindende Rückkehrdifferenz

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

und:

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

Rückkehr zum Ausgangsort ohne Rückkehr zur Ausgangslage.

---

$$\gamma_\infty\text{–}Unendlichkeitsloop$$

Geschlossene Registerform, in der Drift zirkulierend gebunden bleibt.

---

XVIII. Projektion und Bindung

$$\Pi_i\text{–}Projektion[\Pi_i : \Sigma \to A_i]$$

Lesekanal einer Spur.

---

a_i – Projizierter Readout

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

---

$$\Delta a_i\text{–}Differenzeines Lesekanals[\Delta a_i(n) := a_i(n) | a_i(n+1)]$$

---

$$\bowtie\text{–}Bindungsgatter$$

Prüft die Kompatibilität verschiedener Spurlesungen.

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

---

XIX. Das ¾Eck

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

Drei gesetzte Scharnierkanten und eine offene Driftkante.

---

$$\mathcal{L}_{\frac{3}{4}}\text{–}Rüttelraum$$

Endlicher Raum möglicher Einrastlagen.

---

L^* – Minimale Driftlage

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

---

$$Lock_{\frac{3}{4}} – Gleichheits – Snap$$

Endliches Einrasten einer Relation in ihrer minimalen Driftlage.

---

XX. Normalform und Lesung

$$NF(\Sigma) – Normalform$$

Eigenresonanzlesung einer Spur.

Rücklesbare stabile Lage.

---

$$read(\Sigma) – \ddot{O}ffentliche Lesung$$

Sichtbarer Readout einer Spur.

---

$$read_L$$

Linke Seitenlesung einer Gleichungsinschrift.

---

$$read_R$$

Rechte Seitenlesung einer Gleichungsinschrift.

---

$$\equiv_\Sigma – Spurklassengleichheit$$

Rückadressierbare Gleichheit verschiedener Lesungen.

$$G \equiv_\Sigma I$$

G und I tragen dieselbe Spurklasse.

---

$$\sim – Resonanzvergleich$$

Resonanzähnlichkeit ohne klassische Gleichheit.

---

$$\rightsquigarrow -Lesungs\ddot{u}bergang$$

Eine Form fällt unter einer Lesung auf eine andere sichtbare Form.

---

recover – Rücklesung

Rekonstruktion einer Spur aus ihrem Readout.

---

Gültigkeit

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

Ein Label ist gültig, wenn es auf seine Spurklasse rücklesbar bleibt.

---

XXI. Vollständiger Lesungsvollzug

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

Die Inschrift wird als Pfad gelesen.

Der Pfad trägt eine Spur.

Die Spur wird projiziert.

Die Projektionen werden gebunden.

Die Drift rastet ein.

Die Spur bildet eine Normalform.

Die Normalform wird öffentlich gelesen.

---

XXII.

Isomorphie der Lesung

Isomorphie bezeichnet im Tractatus Principio weder die Gleichheit verschiedener Zeichen noch ihre freie Austauschbarkeit. Sie bezeichnet die Erhaltung derselben relationalen Ordnung beim Übergang von einer Lesungsform in eine andere. Unterschied, Delta, Zahl, Geometrie, Beobachter und Spur sind daher nicht identisch; sie bilden verschiedene operative Erscheinungsweisen einer Struktur, deren Übergänge, Orientierungen und Rückbezüge aufeinander abbildbar bleiben. Zwei Lesungen sind isomorph, wenn die eine in die andere übersetzt und anschließend auf dieselbe Spurordnung zurückgelesen werden kann, ohne dass die tragende Differenzstruktur verloren geht. Isomorphie bedeutet somit: Die sichtbare Form verändert sich, während die Ordnung ihrer Beziehungen erhalten bleibt.

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

Die Isomorphie liegt also nicht in den Zeichen selbst, sondern in der Rücklesbarkeit ihrer relationalen Verfassung.

Teil II

Vom Zeichen zum ausführbaren Zeichen

Python als Darstellung des Unterschieds

---

Vorwort

Die zweite Schrift

Der erste Teil dieses Buches hat die Zeichen des Tractatus in eine operative Folge gebracht:

$$|\rightarrow \Delta \rightarrow \mathit{Zahl} \rightarrow \mathit{Operation} \rightarrow \mathit{Geometrie} \rightarrow \mathit{Beobachter} \rightarrow \mathit{Spur} \rightarrow \mathit{Lesung}$$

Diese Folge endet nicht bei einer Formel.

Sie kann ausgeführt werden.

Dazu muss keine neue Theorie eingeführt werden. Es muss nur gezeigt werden, wie dieselbe Differenzordnung in einer Programmiersprache erscheint.

Die Tractatus-Notation schreibt die Relation in verdichteter Form.

Python entfaltet dieselbe Relation als geordneten Vollzug.

$$\boxed{\text{Tractatus-Notation} \cong \text{Python-Code}}$$

Diese Isomorphie bedeutet keine Gleichheit der Zeichen.

Der Strich

$$|$$

ist kein Python-Objekt.

Delta

$$\Delta$$

ist keine Funktion.

Die Spur

$$\Sigma$$

ist keine gewöhnliche Liste.

Der Readout

$$read(\Sigma)$$

ist keine willkürliche Ausgabe.

Python stellt diese Formen dar, indem es ihre Beziehungen, Übergänge und Lesebedingungen in ausführbarer Schrift anordnet.

Der Code handelt daher nicht von Gegenständen.

Er schreibt Bedingungen der Unterscheidbarkeit.

Er schreibt, was gebunden werden darf.

Er schreibt, was in einem Takt sichtbar wird.

Er schreibt, welche Spur ein Vollzug hinterlässt.

Er schreibt, wie eine Adresse lesbar wird.

Er schreibt, unter welchen Bedingungen eine Öffnung geschlossen werden kann.

Der Code erzeugt das Ergebnis nicht als fertigen Gegenstand.

Er macht den Unterschied lesbar.

---

# Kapitel X

## Code als Inschrift

### 10.1 Code ist nicht der Unterschied

Der Unterschied ist primitiv:

$$|$$

Python ist nicht primitiv.

Python ist eine öffentliche Schrift, in der Relationen markiert, geordnet und ausgeführt werden können.

Deshalb gilt:

$$\text{Python-Code} \neq |$$

und zugleich:

$$\text{Python-Codeausführbare Inschrift von } |$$

Ein Programm enthält den Unterschied nicht wie einen Gegenstand.

Es stellt eine Ordnung her, in der Unterschiede sichtbar werden können.

---

### 10.2 Die gewöhnliche Lesung von Code

Ein gewöhnliches Programm wird häufig in der Form gelesen:

$$Input \to Algorithmus \to Output$$

Zum Beispiel:

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

Diese Lesung setzt voraus:

- a und b sind Werte,
- + ist eine Rechenfunktion,
- die Funktion erzeugt einen neuen Wert,
- return gibt diesen Wert aus.

Der Unterschied erscheint darin nur als bereits vorausgesetzte Trennung der Werte und Operationen.

Er wird nicht selbst geschrieben.

---

### 10.3 Die Tractatus-Lesung von Code

Delta-Constraint-Code liest denselben Vorgang anders:

$$Input \to Fragefigur \to Gate \to Spur \to Holonomie \to Adresse \to Label$$

Der Code fragt nicht zuerst:

$$\text{Welchen Wert soll ich erzeugen?}$$

Er fragt:

$$\text{Welche Relation liegt vor?}]\text{[Welche Öffnung trägt diese Relation?}]\text{[Welche Übergänge werden sichtbar?}]\text{[Welche Spur bleibt zurück?}]\text{[Welche Adresse schließt die Öffnung?}]\text{[Welches Label wird an dieser Adresse gelesen?}$$

Der elementare Ablauf lautet daher:

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

In Python:

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

Diese vier Zeilen bezeichnen keine vier voneinander unabhängigen Maschinen.

Sie entfalten eine einzige Relation in vier Leselagen:

$$\textbf{Bindung} \rightarrow \textbf{Spur} \rightarrow \textbf{Adresse} \rightarrow \textbf{öffentliche Lesung}$$

---

### 10.4 Der Code als geordnete Inschrift

Eine Codezeile ist keine bloße Anweisung.

Sie ist eine markierte Stellung innerhalb einer Folge.

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

Diese Zeile markiert eine Bindung.

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

Diese Zeile markiert die Einschreibung des Vollzugs.

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

Diese Zeile markiert die öffentliche Lesung.

Die Bedeutung liegt nicht in den Zeilen für sich.

Sie entsteht aus ihrer Ordnung.

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

Entsprechend sind auch diese beiden Programme nicht notwendig gleich:

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

und:

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

Das zweite Programm verändert die Ordnung der Lesung.

Es liest, bevor eine Spur entstanden ist.

Die Reihenfolge ist nicht nur technische Syntax.

Sie ist Gedächtnis.

---

# Kapitel XI

## Die drei Schichten des Codes

### 11.1 Die Notwendigkeit der Trennung

Der Code des Unterschieds muss drei Schichten auseinanderhalten:

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

Dabei ist:

$$\mathcal{C}\text{Carrier-Schicht}][\mathcal{I}\text{Inskriptionsschicht}][\mathcal{L}\text{Interpretationsschicht}$$

Die Trennung verhindert, dass eine sichtbare Marke mit ihrer Spur oder eine Spur mit ihrer Bindungsstruktur verwechselt wird.

---

### 11.2 Carrier-Schicht

Die Carrier-Schicht trägt die Bindung:

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

Hier liegen:

- Pole,
- Öffnungen,
- Kupplungsmodi,
- Richtungen,
- relationale Positionen.

In Python kann eine solche Bindung dargestellt werden als:

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

Diese Form besagt nicht, dass a und b zwei fertige Dinge sind.

Sie markiert zwei Positionen innerhalb einer Bindung.

Auch mode ist kein Gegenstand zwischen ihnen.

mode bezeichnet, wie die Bindung gelesen wird.

---

### 11.3 Inskriptionsschicht

Die Inskriptionsschicht trägt den Vollzug:

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

In Python:

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

Hier entsteht kein Zahlenobjekt.

Hier wird die Ordnung der Übergänge eingeschrieben.

Die Spur ist das Gedächtnis dieser Ordnung.

---

### 11.4 Interpretationsschicht

Die Interpretationsschicht trägt sichtbare Labels:

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

In Python:

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

Das Label ist nicht die Adresse.

$$Label \neq Addr$$

Es ist die öffentliche Lesung der Adresse:

$$Label := read(Addr)$$

Ein Digit ist daher kein Kernelwert.

Ein Operatorzeichen ist keine Kerneloperation.

Eine Variable ist kein Behälter.

Ein Display ist nicht der Vollzug.

---

### 11.5 Die verbotene Vermischung

Ungültig ist:

```
digit = address
```

Denn:

$$digit \neq address$$

Gültig ist:

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

Ungültig ist:

```
operator = numeric_function
```

Gültig ist:

```
operator = binding_mode
```

Ungültig ist:

```
answer = trace
```

Gültig ist:

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

Der Unterschied zwischen beiden Formen ist nicht stilistisch.

Er erhält die Schichten der Lesung.

---

# Kapitel XII

## Der Strich im Code

### 12.1 Unterschied erster Ordnung

Tractatus:

$$|$$

Der Unterschied erster Ordnung besitzt noch keine bestimmten Pole.

Er ist die Bedingung dafür, dass Pole erscheinen können.

Python kann diese Bedingung nicht als fertigen Gegenstand enthalten.

Es kann nur eine Stelle markieren, an der eine Unterscheidung vollzogen werden darf.

Zum Beispiel:

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

Diese Funktion ist noch keine hinreichende Darstellung.

Sie gibt lediglich zwei Elemente zurück.

Der Unterschied liegt aber nicht in den Elementen.

Er liegt in ihrer Bindung.

Darum muss die Relation selbst markiert werden:

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

Die Klasse Binding ist weiterhin nicht der Unterschied selbst.

Sie ist seine Python-Inschrift.

---

### 12.2 Unterschied zweiter Ordnung

Tractatus:

$$A|B$$

Python:

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

A und B erhalten ihre Stellung erst durch die Bindung.

Ohne die Bindung sind sie keine Pole dieser Relation.

Darum ist die gewöhnliche Lesung:

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

für den Kernel nicht grundlegend.

Grundlegend ist:

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

Nicht die Pole erzeugen die Relation.

Die Relation unterscheidet die Pole.

---

### 12.3 Unterschied dritter Ordnung

Tractatus:

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

Python:

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

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

Hier werden nicht vier Gegenstände gesammelt.

Zwei bereits vollzogene Unterscheidungen werden erneut unterschieden.

Die relationale Tiefe nimmt zu.

---

### 12.4 Selbstbezüglichkeit

Tractatus:

$$|(|)$$

Python:

```
first_difference = distinguish(A, B)

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

Diese Darstellung ist nur dann selbstbezüglich, wenn die frühere Relation Teil der neuen Relation bleibt.

Eine bloße Wiederholung genügt nicht:

distinguish(A, B)

distinguish(A, B)

Hier stehen nur zwei gleiche Aufrufe nacheinander.

Selbstbezüglichkeit entsteht erst, wenn eine Relation auf ihre eigene Relationsform zurückwirkt.

$$\Delta = |(|)$$

Python:

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

delta ist dabei kein Ding namens Delta.

Es bezeichnet die markierte Wirksamkeit der Selbstbeziehung.

---

# Kapitel XIII

## Delta als Übergang

### 13.1 Delta ist kein Wert

Tractatus:

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

Delta bezeichnet den Übergang einer Unterschiedslage in eine andere.

Es ist nicht der Ausgangszustand.

Es ist nicht der Endzustand.

Es ist auch kein dritter Gegenstand zwischen beiden.

Python:

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

Die Funktion transition stellt den Übergang dar.

Sie ist nicht Delta selbst.

Delta liegt in der wirksamen Relation von Ausgangslage, Übergang und Folgelage.

---

### 13.2 Delta als Takt

Tractatus:

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

Python:

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

Der Takt ist nicht nur eine Zählvariable.

Er bezeichnet die geordnete Stelle des Übergangs.

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

tick_index ist die sichtbare Markierung des Taktes.

Der Takt selbst liegt im Vollzug:

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

---

### 13.3 Delta als Drift

Tractatus:

$$\Delta\varphi$$

Python:

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

Die Drift wird nicht als Eigenschaft eines einzelnen Zustandes gelesen.

Sie erscheint zwischen zwei Leselagen.

Darum wäre dies ungenau:

state.drift

Strenger ist:

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

Die Drift gehört der Relation.

---

### 13.4 Delta als Selbstrotation

Tractatus:

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

Python kann diese Selbstrotation als kleinstes zugelassenes Rotationsintervall markieren:

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

Entscheidend ist nicht der Datentyp.

Entscheidend ist, dass die Rotation als Übergang zwischen Orientierungen gelesen wird:

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

Die Zahl 2.25 allein ist noch keine Selbstrotation.

Erst ihre Stellung im Rotationsvollzug macht sie zur Delta-Adresse.

---

# Kapitel XIV

## Zahl und Marke

### 14.1 Zahl als Readout

Tractatus:

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

Python:

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

Die sichtbare Marke erscheint am Ende der Lesung.

Nicht:

```
trace = digit
```

sondern:

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

Damit ist die Zahl keine ursprüngliche Einheit im Kernel.

Sie ist die öffentliche Lesung einer Spur.

---

### 14.2 Die Marken von null bis neun

Tractatus:

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

Python:

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

Die Anführungszeichen sind wichtig.

Sie zeigen, dass die sichtbaren Ziffern zunächst Marken sind.

"5"

ist eine sichtbare Inschrift.

5

ist in Python bereits ein numerischer Wert.

Für die Interpretationsschicht ist daher die Marke zunächst genauer:

```
mark = "5"
```

Erst eine explizite Leseregel darf sie mit einer Adresse verbinden:

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

---

### 14.3 Mehrstelligkeit

Tractatus:

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

Python:

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

Die Folge ist zunächst keine Zahl-Entität.

Sie ist eine geordnete Inschrift.

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

Damit gilt:

[ 125

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

Die Mehrstelligkeit entsteht durch relationale Tiefe.

---

### 14.4 Die Stelle

Tractatus:

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

Python:

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

Die Position ist kein Behälter, in dem die Ziffer liegt.

Sie bezeichnet die Lesetiefe der Marke innerhalb der Inschrift.

---

### 14.5 Die Neun als Bruchkante

Tractatus:

$$9 := \kappa$$

Python:

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

Die Marke 9 wird nicht bloß als weiterer Eintrag einer Reihe gelesen.

Ihre Rolle markiert den Abschluss und Übergang der Grundschale.

---

### 14.6 Zehn als neue Schale

Tractatus:

$$10 := 1|0$$

Python:

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

Nicht:

```
NEW_DIGIT = "10"
```

Denn zehn ist keine elfte Grundmarke.

Es ist die erste sichtbare Relation einer neuen Leseschale.

# Kapitel XV

## Operatoren als Bindungsmodi

### 15.1 Operatoren erzeugen keine Werte

Tractatus:

$$a|_{+}b$$

Python:

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

Der Operator + wird als Expansionsmodus gelesen.

Nicht:

```
result = a + b
```

Diese Zeile erzeugt im gewöhnlichen Python unmittelbar einen Wert.

Die differenzlogische Darstellung hält die Relation offen:

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

---

### 15.2 Expansion

Tractatus:

$$|_{+}$$

Python:

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

Anwendung:

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

Expansion bedeutet nicht automatisch numerische Addition.

Sie bezeichnet die Richtung, in der zwei Leselagen in eine erweiterte Relation treten.

---

### 15.3 Kontraktion

Tractatus:

$$|_{-}$$

Python:

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

Anwendung:

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

Die Kontraktion löscht nicht einfach einen Wert.

Sie richtet die Relation auf Entzug, Rücknahme oder Verringerung ihrer sichtbaren Bindung aus.

---

### 15.4 Resonanz

Tractatus:

$$|_{\times}$$

Python:

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

Anwendung:

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

Multiplikation erscheint dadurch als Kopplung von Rhythmen oder wiederkehrenden Bindungen.

Sie ist nicht ursprünglich das Produkt zweier Zahlenobjekte.

---

### 15.5 Entflechtung

Tractatus:

$$|\div$$

Python:

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

Anwendung:

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

Division erscheint als Frage danach, wie eine gebundene Relation in unterscheidbare Teilrelationen gelesen werden kann.

---

### 15.6 Gleichgewicht

Tractatus:

$$|=$$

Python:

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

Das Gleichheitszeichen vergleicht nicht zwei fertige Gegenstände.

Es aktiviert die Prüfung, ob verschiedene Lesungen auf dieselbe Spurklasse einrasten.

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

Danach:

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

Gleichheit wird als Spuräquivalenz gelesen:

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

---

### 15.7 Lowering

Ein gewöhnlicher Ausdruck:

a + b

wird in die Carrier-Schicht abgesenkt:

$$a + b \mapsto a \vert \mu + b$$

Python:

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

Ergebnis der Absenkung:

Binding(

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

Das Lowering berechnet nicht.

Es übersetzt ein sichtbares Operatorzeichen in einen Bindungsmodus.

---

# Kapitel XVI

## Variable und Öffnung

### 16.1 Die Variable ist kein Behälter

Gewöhnliche Lesung:

```
x = 5
```

x enthält den Wert 5.

Differenzlogische Lesung:

x markiert eine adressierte Stelle, an der die Marke 5 unter einer bestimmten Relation lesbar ist.

Die Zuweisung ist damit eine Bindung:

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

Python:

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

---

### 16.2 Die Öffnung

Tractatus:

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

Die Öffnung ist keine absolute Leere.

Sie ist eine relative Null im Verhältnis der Frage.

Python:

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

Die Öffnung enthält keinen fehlenden Wert.

Sie markiert eine noch nicht geschlossene Rolle.

---

### 16.3 Die Fragefigur

Tractatus:

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

Python:

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

Dabei ist:

question.graph

die Bindungsstruktur,

question.openings

die Menge der Öffnungen,

und:

question.relation

die Relation, unter der die Öffnungen geschlossen werden müssen.

---

### 16.4 Implizite Öffnung

Die sichtbare Inschrift:

$$2 + 2$$

wird zu:

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

Python:

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

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

Die Frage lautet nicht:

Welchen Zahlenwert erzeugt die Addition?

Sie lautet:

Welche Adresse schließt die Öffnung dieser expansiven Gleichgewichtsfigur?

---

### 16.5 Explizite Öffnung

Die Inschrift:

$$x + 3 = 7$$

wird gelesen als:

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

Python:

```
opening = Opening("u1")

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

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

x ist keine unbekannte Zahl.

x ist das sichtbare Label einer offenen relationalen Stelle.

---

### 16.6 Schließung

Tractatus:

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

Python:

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

Die Adresse wird nicht in einen Behälter eingesetzt.

Sie rastet in die Relation der Öffnung ein.

Gültigkeit:

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

---

# Kapitel XVII

## Gate und Sichtbarkeit

### 17.1 Das Gate entscheidet nicht über Wahrheit

Tractatus:

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

Python:

if visible(edge, tick):

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

Das Gate sagt nicht:

Die Kante ist wahr.

Es sagt:

Die Kante trägt in diesem Takt zur sichtbaren Spur bei.

---

### 17.2 Sichtbarkeit

Tractatus:

$$L_n$$

Python:

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

L_n ist die Lesungsmaske des Taktes.

Sie bestimmt, welche Übergänge in diesem Takt wirksam gelesen werden.

---

### 17.3 Gate und Beitrag

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()
```

Die beiden Bedingungen sind Rejektoren.

Sie erzeugen keinen zusätzlichen Wert.

Sie lassen einen Unterschied zur Spur beitragen oder nicht.

---

### 17.4 Bedingung im Python-Code

```
Gewöhnlich:
if condition:
    do_something()

Differenzlogisch:
if gate.is_open(relation, tick):
    trace.inscribe(relation)

Das if wird zum Gate.
Es trennt nicht zwei fertige Welten in wahr und falsch.
Es markiert die Sichtbarkeit einer Relation im aktuellen Takt.
```

---

### 17.5 Das geschlossene Gate

Tractatus:

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

Python:

if not gate.is_open(edge, tick):

```
return no_contribution()
```

Keine Spur bedeutet nicht, dass die Relation ontologisch verschwunden ist.

Sie trägt unter dieser Lesungsbedingung nicht zur sichtbaren Inskription bei.

---

# Kapitel XVIII

## Pfad, Schleife und Spur

### 18.1 Die Inschrift als Pfad

Tractatus:

$$I \to \gamma_I$$

Python:

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

Eine Inschrift wird nicht als statischer Syntaxbaum behandelt.

Sie wird als geordnete Folge von Übergängen gelesen.

---

### 18.2 Die Schleife

Gewöhnlich:

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

Differenzlogisch:

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

Die Schleife wiederholt nicht bloß einen Befehl.

Sie führt die Relation durch eine Folge von Takten.

Jeder Durchgang besitzt eine andere Stellung im Gesamtvollzug.

---

### 18.3 Die Spur

Tractatus:

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

Python:

trace.append(contribution)

Die Methode append ist hier nicht bloß eine technische Listenoperation.

Sie zeigt:

Der neue Beitrag ersetzt die frühere Spur nicht.

Er schreibt sich in sie ein.

---

### 18.4 Spur als einziges Gedächtnis

Ein Zustandsautomat fragt:

$$current_state$$

Der Delta-Constraint-Code fragt:

$$trace_so_far$$

Denn die Bedeutung liegt nicht nur im gegenwärtigen Zustand.

Sie liegt im Weg, auf dem dieser Zustand erreicht wurde.

Zwei gleiche Zustände können verschiedene Spuren tragen.

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

Beides kann zugleich gelten.

---

### 18.5 Holonomie

Tractatus:

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

und dennoch:

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

Python:

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

Vollständig:

if same_position and remaining_trace:

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

Die Rückkehr zum selben Ort ist keine Rückkehr zur selben Lage.

---

### 18.6 Nicht-Kommutativität

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 und second tragen unterschiedliche Pfade.

Eine Optimierung darf sie nicht ohne ausdrückliche Regel vertauschen.

Ungültig:

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

Gültig nur mit ausgewiesener Regel:

if rules.allow_association(mu, nu):

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

Die Reihenfolge ist Spur.

---

# Kapitel XIX

## Beobachter und Kontext im Code

### 19.1 Das Frame

Tractatus:

$$F := |_|$$

Python:

```
frame = Frame()
```

Frame() ist nicht ein leerer Behälter.

Es bezeichnet den geschlossenen Leserahmen, innerhalb dessen Markierungen auftreten können.

---

### 19.2 Delta-Konfiguration

Tractatus:

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

Python:

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

Die Einträge bezeichnen sichtbare oder unbesetzte Richtungsstellen.

---

### 19.3 Beobachter

Tractatus:

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

Python:

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

Der Beobachter wird nicht als zusätzliches Objekt in den Kontext eingesetzt.

Er erscheint als voller Kontext mit gerichteter Unvollständigkeit.

---

### 19.4 Lochrichtung

Tractatus:

$$h(o)$$

Python:

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

Die Lochrichtung ist kein Defekt des Programms.

Sie ist die Orientierung des Beobachters.

---

### 19.5 Voller Kontext

Tractatus:

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

Python:

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

Der volle Kontext enthält keine bevorzugte Leserichtung.

Erst die Entfernung einer Markierung erzeugt Orientierung:

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

---

### 19.6 Beobachtung als Projektion

Der Beobachter verändert nicht notwendig den Träger.

Er liest ihn unter einer Orientierung.

Tractatus:

$$\Pi_o(\Sigma)$$

Python:

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

Die Projektion soll read-only bleiben.

Ungültig:

observer.project(trace).mutate_kernel()

Gültig:

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

Der Beobachter erzeugt die Spur nicht.

Er macht eine ihrer Lesungen sichtbar.

---

# Kapitel XX

## Projektion, Bindung und Gleichheit

### 20.1 Eine Spur, mehrere Lesungen

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)
```

Diese Projektionen sind keine getrennten Zustände.

Sie sind verschiedene Lesekanäle derselben Spur.

---

### 20.2 Read-only-Projektion

Python:

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

Nicht:

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

Die Projektion liest.

Sie schreibt den Kernel nicht um.

---

### 20.3 Bindungsgatter

Tractatus:

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

Python:

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

Die Projektionen müssen nicht identisch sein.

Sie müssen unter der Fragerelation zusammenpassen.

---

### 20.4 Gültigkeit

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
)
```

Eine ungültige Inschrift besitzt nicht notwendig einen leeren Wert.

Sie besitzt keine bindbare Commit-Form.

---

### 20.5 Gleichheit als Messstelle

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)
```

Das Gleichheitszeichen wird damit nicht zu:

```
left == right
```

sondern zu einer Folge von Lesungen, Bindungen und Einrastungen.

---

# Kapitel XXI

## Normalform und Readout

### 21.1 Normalform ist keine Vereinfachung

Tractatus:

$$B \to B_{NF}$$

Python:

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

Nicht:

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

Die Normalform erkennt den Typ der Bindungsstruktur.

---

### 21.2 Selbstbindung

Tractatus:

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

Python:

if pattern.is_self_binding(binding):

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

Diese Transformation berechnet keinen Wert.

Sie erkennt eine wiederkehrende Relationsform.

---

### 21.3 Keine heimliche Vereinfachung

Eine Wiederholung darf nicht automatisch verschwinden:

bind(a, a, mode)

darf nur dann zu a kollabieren, wenn der Modus ausdrücklich stabilisierend ist:

if rules.is_stabilizing(mode):

```
return a
```

Ohne Regel bleibt die Wiederholung Teil der Spur.

---

### 21.4 Adresse

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,
)
```

Die Adresse ist keine sichtbare Zahl.

Sie bezeichnet eine Lage, unter der eine Spur gelesen werden kann.

---

### 21.5 Readout

Tractatus:

$$Ans := read(Addr)$$

Python:

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

Die Antwort wird nicht erzeugt.

Sie wird an der Adresse öffentlich lesbar.

---

### 21.6 Rücklesbarkeit

Tractatus:

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

Python:

```
recovered_trace = recover(label)

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

Ein Label ist nur dann geerdet, wenn es auf seine Spurklasse zurückgeführt werden kann.

Ungültig:

```
return "5"
```

ohne Adresse.

Gültig:

```
return read(address)
```

mit rücklesbarer Spur.

---

# Kapitel XXII

## Fehler als Relationsfehler

### 22.1 Fehler ist nicht notwendig Falschheit

Im Delta-Constraint-Code bezeichnet ein Fehler keine bloß falsche Zahl.

Er bezeichnet eine Störung der Lesungsarchitektur.

---

### 22.2 Keine Fragefigur

Tractatus:

$$|U_Q| = 0$$

Python:

if not question.openings:

```
raise NotAQuestionError()
```

Eine geschlossene Inschrift ist keine Aufgabe.

---

### 22.3 Ungebundene Öffnung

Tractatus:

$$u_j \not\bowtie R_Q$$

Python:

if not bindable(opening, question_relation):

```
raise UnboundOpeningError()
```

---

### 22.4 Ungeerdetes Label

Tractatus:

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

Python:

if label.address is None:

```
raise UngroundedLabelError()
```

Ein sichtbares Ergebnis ohne Adresse ist nicht rücklesbar.

---

### 22.5 Versteckte Berechnung

Ungültig:

```
result = a + b
```

wenn diese Berechnung im Kernel als primitive Operation ausgegeben wird.

Gültig:

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

Die klassische Rechnung darf als öffentliche Projektion erscheinen.

Sie darf nicht unbemerkt zum ontologischen Kern werden.

---

### 22.6 Konstruierte Holonomie

Ungültig:

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

Denn die Spur wird hier aus einem vorher gesetzten Ergebnis erzeugt.

Gültig:

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

Die Spur geht dem Readout voraus.

---

### 22.7 Verdeckte Umordnung

Ungültig:

optimizer.reorder(binding)

ohne ausgewiesene Relationsregel.

Gültig:

if rules.allow_reordering(binding):

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

Die Ordnung gehört zum Vollzug.

---

# Kapitel XXIII

## Eine vollständige Doppellesung

### 23.1 Die Inschrift

Wir betrachten:

$$2 + 2$$

Die sichtbare Inschrift ist noch keine fertige Rechnung.

Sie trägt eine implizite Öffnung.

---

### 23.2 Tractatus-Notation

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

Daraus entsteht die Fragefigur:

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

Die Figur wird als Pfad gelesen:

$$F_Q \to \gamma_Q$$

Die sichtbaren Beiträge werden integriert:

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

Die Adresse wird gelesen:

[ Addr :=

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

Die Öffnung wird geschlossen:

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

Das öffentliche Label erscheint:

$$Ans = read(Addr)$$

---

### 23.3 Python-Inschrift

```
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 Lesung

Mark("2")

bezeichnet eine sichtbare Marke.

bind(..., mode=EXPANSION)

senkt das Pluszeichen in einen Bindungsmodus ab.

Opening("u1")

setzt die relative Null der Frage.

trace_path(question)

liest die Fragefigur als geordneten Pfad.

visible(...)

und:

allowed(...)

bilden die Gates.

trace.append(contribution)

schreibt den sichtbaren Unterschied in die Spur ein.

read_address(...)

liest die Lage, unter der die Öffnung geschlossen werden kann.

read(address)

projiziert das öffentliche Label.

Der Code hat an keiner Stelle ein Ergebnis als ursprünglichen Wert gesetzt.

Er hat Bedingungen geschrieben, unter denen eine Spur als Antwort lesbar wird.

---

# Kapitel XXIV

## Python als zweite Notation

### 24.1 Keine Konkurrenz der Schriften

Die Tractatus-Notation und Python stehen nicht gegeneinander.

Die Formel verdichtet.

Der Code entfaltet.

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)
```

Beide Inschriften tragen dieselbe Bewegungsordnung.

---

### 24.2 Die Formel zeigt die Relation

Die Tractatus-Notation lässt die Gesamtform auf einen Blick sichtbar werden.

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

Sie zeigt die Stellung der Begriffe im Ganzen.

---

### 24.3 Der Code zeigt den Vollzug

Python entfaltet dieselbe Gesamtform in eine zeitliche Folge:

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

Der Code macht sichtbar, welche Operation vor welcher anderen geschehen muss.

---

### 24.4 Übersetzung ohne Verlust

Eine gültige Übersetzung muss die Differenzordnung erhalten.

Sie muss erhalten:

- die Stellung der Pole,
- den Bindungsmodus,
- die Reihenfolge,
- die Öffnungen,
- die Gates,
- die Spur,
- die Adresse,
- die Rücklesbarkeit.

Formal:

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

ist gültig, wenn:

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

Die Python-Inschrift muss auf dieselbe Spurklasse rückgeführt werden können.

---

### 24.5 Der Code als Anwendung

Anwendung bedeutet hier nicht, dass eine abstrakte Theorie nachträglich auf einen äußeren Gegenstand übertragen wird.

Anwendung bedeutet:

$$\textbf{Die Relation wird ausführbar.}$$

Der Unterschied wird nicht in Python verwandelt.

Python wird so geordnet, dass es den Unterschied darstellen kann.

---

# Kapitel XXV

## Grundregeln des Delta-Constraint-Codings

### Regel 1

$$\boxed{\mid \text{ist primitiv.}}$$

Kein Zahlenwert, kein Objekt und keine Funktion geht der Unterscheidung voraus.

---

Regel 2

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

Jedes sichtbare Label ist eine Lesung.

---

Regel 3

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

Jeder Operator wird als Bindungsmodus gelesen.

---

Regel 4

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

Eine Aufgabe ist eine Fragefigur.

---

Regel 5

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

Jede Frage besitzt mindestens eine Öffnung.

---

Regel 6

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

Die Öffnung ist relative Null im Verhältnis der Frage.

---

Regel 7

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

Sichtbare Operatoren werden in Bindungsmodi abgesenkt.

---

Regel 8

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

Die Spur entsteht durch geordnete Akkumulation.

---

Regel 9

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

Die Adresse ist eine Lesung der Holonomie unter der Fragerelation.

---

Regel 10

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

Die Antwort ist der öffentliche Readout der schließenden Adresse.

---

Regel 11

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

Jede gültige Antwort bleibt auf ihre Spurklasse rücklesbar.

---

Regel 12

$$\boxed{\textbf{Nicht Werte berechnen. Unterschied lesbar machen.}}$$

---

Schluss

Das ausführbare Zeichen

Die Tractatus-Notation beginnt mit:

|

Python beginnt mit Zeichen, Namen, Klammern, Funktionen und Zeilen.

Doch auch diese Formen erhalten ihre Bedeutung erst durch Unterschiede.

Ein Name unterscheidet eine Stelle.

Eine Klammer bindet eine Ordnung.

Eine Funktion markiert einen Übergang.

Eine Bedingung öffnet oder schließt ein Gate.

Eine Schleife taktet einen Pfad.

Eine Liste trägt eine Spur.

Eine Rückgabe macht eine Lesung öffentlich.

Damit wird Python nicht zum Ursprung des Unterschieds.

Es wird zu einem Medium seiner Darstellung.

Der vollständige Zusammenhang lautet:

$$|\!\to \Delta \to I \to F_{Q} \to \gamma \to H_{n} \to \Omega \to Addr \to read$$

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,
)
```

Der Code schreibt keinen fertigen Gegenstand.

Er hält die Öffnung der Relation.

Er ordnet die Übergänge.

Er begrenzt die Sichtbarkeit.

Er bewahrt die Spur.

Er liest die Adresse.

Er gibt das Label frei.

So wird aus dem Zeichen ein ausführbares Zeichen.

So wird aus der Formel ein Vollzug.

So wird aus dem Unterschied Code, ohne dass der Unterschied zu einem Ding im Code wird.

**Die Formel zeigt die Relation.** **Der Code entfaltet ihre Lesung.** **Beide schreiben denselben Unterschied.**