Paper im Rahmen des Tractatus Principio |

—

UXP - Why Compression Is a Dead End

von

C.H.F. Warnke

UXP

Unit Exchange Parameter

Warum Kompression eine Sackgasse ist

Der UXP Archiver – Readadressierung von Information durch Differenz C.H.F. Warnke The Future Foundation Einstiegspapier Nr. 03 Version 3.7 – Streaming-Vollzug / Schnittstellen-Projektionsausgabe – August 2026 https://the-future-foundation.com

Zusammenfassung

Kompression funktioniert. Das ist nicht das Problem.

Das Problem ist die Frage, die Kompression bereits akzeptiert hat, bevor sie beginnt: Es gibt ein fertiges Objekt namens Daten, das Objekt enthält Information, und die Aufgabe besteht darin, dieses Objekt zu bewahren, während man weniger Zeichen verwendet, um es darzustellen. Bessere Wörterbücher, Entropiemodelle, Transformationen und Prädiktoren können das Ergebnis dramatisch verbessern. Sie ändern diesen Ausgangs punkt nicht.

UXP setzt eine Operation früher an.

UXP steht für Unit Exchange Parameter. Die sichtbare Einheit der Darstellung kann sich ändern, während die adressierte Relation das bleibt, was wiederherstellbar sein muss. Binär-, Dezimal-, Hexadezimal- und Byte-Notation sind daher öffentliche Oberflächen der Relation. Sie sind keine verschiedenen Arten von Information.

Der Tractatus Principio setzt Differenz vor Zahl:

|

Ihre relationale Entfaltung wird geschrieben als

$A|B$, und ihre Selbstreferenz ist

$$\Delta := |(|)$$

Dies sind keine zeitlichen Stadien. Sie sind gleichzeitige Ordnungen derselben Unterscheidung. Zahl ist ein Auslesen stabilisierter relationaler Ordnung, nicht das primitive Objekt, aus dem die Relation gebaut wird.

Der UXP Archiver wendet diese Typendisziplin auf einer digitalen Ausführungsoberfläche an. Eine digitale Datei ist bereits als geordnete endliche numerische Relation lesbar, oder als eine digitale Inschrift, die zu einer solchen numerischen Relation isomorph ist. Die numerischen Zeichen sind der Differenz nicht ontologisch vorgängig; sie sind das maschinelle Auslesen, unter dem die digitale Inschrift ausführbar ist. Die Inschrift wird als Pfad gelesen, der Pfad als Spur, und die Spur wird unter einem festen Ausführungsprofil projiziert, gebunden, geschlossen und normalisiert.

Bevor die große Kaskade erscheint, zeigt die Zahl bereits den operativen Rhythmus derselben Selbstreferenz.

Der Tractatus schreibt:

$$\Delta := |(|)$$

Band VIII liest dasselbe Δ als Takt, Übergang, Markierung, Drift und Selbstrotation. Dies sind keine verschiedenen Delta-Objekte. Sie sind verschiedene operative Lesarten derselben selbstreferenziellen Unterscheidung.

Sei:

Tj (Δ) die konkrete geordnete Erscheinung desselben Δ an einer lesbaren Übergangsposition bezeichnen. Dann kann ein numerischer Rhythmus schematisch geschrieben werden als:

$$\Delta[n] := T1\ (\Delta)|T2\ (\Delta)|\ \cdots\ |Tn\ (\Delta)$$

Die entscheidende Relation ist:

$$\Delta[n] \equiv\Delta\ \Delta$$

Dies löscht weder die Ordnung noch die Anzahl der vollzogenen Übergänge. Es besagt, dass der wiederholte Takt keine neuen Klassen von Delta erzeugt. Jeder lesbare Schritt bleibt dieselbe Selbstreferenz der Differenz an einer anderen geordneten Position.

Band I beschreibt die numerische Konsequenz von der anderen Seite. Die 1 beginnt, sich auf sich selbst zu beziehen, und die 2 ist die sichtbare Umschreibung dieser Eigenrelation. Dasselbe Kapitel hält die 1 vierfach in der 4 aktiv:

$$1|1|1|1$$

UXP liest die Ziffer daher als öffentliches Takt-Auslesen:

$n \coloneqq \mathit{read}\mathrm{num}\ (\Delta[n]\ )$.

Das Symbol 2 ist keine zweite primitive numerische Substanz. Es ist die öffentliche Umschreibung zweier geordneter Lesungen derselben selbstrelationalen Differenz. Ebenso enthält 4 nicht vier gespeicherte Einheiten; sie macht denselben Unterscheidungsrhythmus an vier Positionen lesbar. So erscheint ein Ziffernsystem.

Verschiedene stabilisierte numerische Symbole werden auf einer gemeinsamen öffentlichen Takt-Ebene synchronisiert: (*r*)

$$\mathscr{N}10 = \mathit{readnum}(\mathit{Tr}, 0|\mathit{Tr}, 1|\cdots|\mathit{Tr}, 9) = 0|1|2|3|4|5|6|7|8|9$$

Der Index r bezeichnet die Referenzschale, nicht eine Eigenschaft des einzelnen Glyphen. Somit erwerben 2, 4 und 8 nicht unterschiedliche Adresstiefen, nur weil ihre internen geometrischen oder selbstrelationalen Lesarten verschieden sind. Diese qualitativen Lesarten bleiben als andere

Kanäle derselben Spur verfügbar. Auf der Ziffernebene wurden sie auf eine gemeinsame Schale umgeschrieben.

Die 9 bleibt:

$9 := \kappa$, die sichtbare Bruchkante dieser Schale.

Aber κ ist nicht Tiefe.

Die nächste öffentliche Form:

$10 := 1|0$ fügt keine elfte primitive Ziffernposition hinzu. Sie markiert, dass die bereits differenzierte Taktordnung selbst zur Referenz einer anderen Unterscheidung geworden ist. Tiefe beginnt daher dort, wo eine vollendete selbstunterscheidende Ordnung selbst wieder unterschieden wird. Dies gibt der UXP-Kaskade ihre exakte numerische Form: nicht Zahl-Objekt nach Zahl-Objekt, sondern derselbe Delta-Takt, gelesen, geschlossen, readaddressed und auf einer anderen Referenzebene wiedergelesen. Für eine feste öffentliche Position m bleiben neun Gegenpositionen auf derselben Zehn-Positionen-Schale. Dies ist ein Anwendungslemma: Die Fixierung einer Position in der vollständigen dezimalen Takt-Ebene lässt neun andere relationale Positionen übrig. Ihre Ordnung ist die Ordnung des gemeinsamen Takts, nicht eine willkürliche invertierbare Permutation.

Die zentrale operative Proposition lautet dann: Hier bedeutet „Komplexität“ keine Akkumulation verborgener Informationsobjekte. Sie bedeutet die gleichzeitige Bindung mehr kompatibler relationaler Bedingungen. Unter Bedingungskomposition,

$Allow\Gamma 1 \otimes \Gamma 2 = Allow\Gamma 1 \wedge Allow\Gamma 2$, und daher

$$\delta(x;\ \Gamma 1 \otimes \Gamma 2\ ) \subseteq \delta(x;\ \Gamma 1\ )$$

Eine reichere Relation kann die zulässige Fortsetzungsmenge nur bewahren oder verengen. Wenn die aktuelle Leseoberfläche unzureichend unterscheidend bleibt, identifiziert UXP unterschiedliche Relationen nicht. Es ändert die Referenzebene. In der Dezimalschale ist 10 kein weiterer Schritt auf derselben Oberfläche; sie ist eine neue Relation der Tiefe. Die Adresse einer Tiefe wird zur Inschrift der nächsten. Entscheidend ist, dass die Kaskade keine Folge sichtbarer dezimaler Reduktionen ist. Keine Zwischenschale wird durch einen zehnstelligen Wert ersetzt. Jede neue Schale wird aus dem bereits etablierten Pfad, der Spur und der transportierten Bedingungsordnung der vorhergehenden Schale adressiert. Die nächste Ebene ist daher ein Readaddressing der bereits erzeugten Relation, kein Neuanfang von ihrer sichtbaren Projektion. Schematisch: (2)

(1)

$$I\ (0) \to \gamma\ (0) \to \Sigma(0) \to AdrP \to \gamma\ (1) \to \Sigma(1) \to AdrP \to \cdots \to \Sigma(*)$$

Bei jedem Übergang ist die neue Adresse in der vorhergehenden Spur und in der Bedingungs ordnung gegründet, die durch dasselbe feste Profil transportiert wird. Die öffentliche Dezimalform erscheint erst, nachdem diese rekursive Schalenrelation geschlossen ist.

Das öffentliche Ergebnis ist daher keine willkürliche Teilprojektion. Es ist ein vollständiges Normalform- Auslesen:

$LNF = readNF\ \boldsymbol{NF\varphi(\Sigma)}$.

Für das öffentliche Zehn-Positionen-Profil bindet der finale Schluss die relevanten Lesekanäle derselben Spur, bevor das öffentliche Auslesen genommen wird. Relationale Tiefe und Adresse sind nicht zwei Nutzlaststücke, die in einen äußeren Container gelegt werden. Sie sind Lesungen der bereits integrierten Spur:

$\ell I := readd\ \boldsymbol{\Sigma(\gamma I)}$,

$AdrI := readAdr\ \boldsymbol{\Sigma(\gamma I)}$.

Die Notation für ihre Bindung bezeichnet daher Kompatibilität innerhalb einer spur-gegründeten Normalform, nicht das Verpacken zweier wiedergewonnener Felder. Gegen-Lesung „entpackt“ Tiefe und Adresse nicht aus dem Dezimallabel. Sie führt zuerst die vollständige öffentliche Normalform-Lesung in ihre Spurklasse zurück:

$LNF \rightsquigarrow N F\varphi \rightsquigarrow [\Sigma]\equiv\Sigma$.

Tiefe, Adresse, Schalenposition und Orientierung sind dann aus dieser Spurklasse unter demselben festen Profil wieder lesbar.

Weder Tiefe noch Pfad wird als Nebenobjekt gespeichert. Terminierung muss ebenso eine Schluss bedingung derselben Relation sein, keine extern gelieferte Quelllänge.

Der .uxp-Körper ist leer. Kein Quellblock, keine Pfadliste, keine Stück-Karte, keine zweite Rekonstruktionsroute, kein persistenter Knotengraph und keine Anwendungslage pro Datei darf eine fehlende Unterscheidung liefern. Diese Architektur ist exakt nur auf dem profil-zulässigen Bereich, für den das öffentliche Normalform-Label eine Spurklasse unter der festen Regel identifiziert. Das Zehn-Positionen-Profil entgeht daher seiner eigenen öffentlichen Domänenbedingung nicht: Exaktheit für beliebige Dateien ist ein separater Abdeckungsanspruch und muss durch dieselbe gegen-lesbare Relation von der Quell inschrift zur öffentlichen Normalform demonstriert werden, nicht aus der bloßen Existenz der Kaskade abgeleitet. UXP ist keine Kompression.

Kompression verändert den Körper, der eine Darstellung trägt.

UXP verändert die Lesung, unter der eine Relation öffentlich wird.

1 Kompression beginnt zu spät

Bevor das Argument beginnt, muss der Name klar sein.

```
UXP = Unit Exchange Parameter. Der Ausdruck benennt keine neue Art von Zahl-Objekt.
Er benennt die operative Tatsache, dass eine sichtbare Darstellungseinheit
ausgetauscht werden kann, während die adressierte Relation das bleibt, was für das
```

```
Zurücklesen bewahrt werden muss. Eine Dezimalziffer, ein hexadezimales Zeichen und
ein Byte sind daher verschiedene öffentliche Einheiten der Inschrift, nicht
verschiedene Fundamente der Relation. Deshalb kann die UXP-Referenz den öffentlichen
Radix ändern, ohne Radix als Ontologie zu behandeln. Das Wort „Zahl“ wird dennoch in
diesem Text durchgehend erscheinen, weil die exakte öffentliche Adresse des Archivers
eine Ganzzahl ist. Das kehrt die Tractatus- Ordnung nicht um. Die Zahl ist das
lesbare Ergebnis der Relation. Sie ist nicht die primitive Quelle, aus
```

der die Relation gebaut wird. Kompression ist einer der großen ingenieurtechnischen Erfolge der Informatik. Nichts in diesem Text erfordert, das zu leugnen. Der Titel „Warum Kompression eine Sack gasse ist“ ist keine Vorhersage, dass ZIP, DEFLATE, LZ-Familienverfahren, arithmetisches Kodieren, Transformationskodierung oder künftige Kompressoren plötzlich aufhören werden, nützlich zu sein. Die Sackgasse ist erklärend, nicht praktisch.

Kompression beginnt, nachdem eine bestimmte Ontologie von Daten bereits akzeptiert worden ist. Es gibt ein Objekt: X.

Es enthält einen endlichen Informationsgehalt. Ein Kompressor wendet eine Transformation C an und erzeugt eine zweite Darstellung: X –C–> Y mit der praktischen Hoffnung, dass size$Y$ < size$X$ während irgendein inverses Verfahren X wiederherstellt. Dieses ingenieurtechnische Problem ist legitim. Aber beachte, was bereits fixiert ist, bevor C erscheint. X ist bereits eine Einheit. Sein Anfang und Ende sind bekannt. Seine Zeichen sind unterscheidbar. Ihre Ordnung ist lesbar. Die Maschine weiß bereits, was als erstes Byte, nächstes Byte und letztes Byte zählt. Der Kompressionsalgorithmus erzeugt diese Unterscheidungen nicht. Er empfängt sie.

UXP beginnt dort. Die Frage lautet nicht zuerst: Wie kann X kleiner gemacht werden?

Sie lautet: Was machte X als X lesbar?

Das ist dieselbe fundamentale Umkehrung, die im Einstiegspapier Nr. 01 eingeführt wurde. Ein mathematisches Objekt ist der Unterscheidung, die erforderlich ist, um es zu identifizieren, nicht operativ vorgängig. Der vorliegende Text trägt diese Umkehrung in ausführbare Darstellung. Die Bewegung ist einfach auszusprechen und leicht zu übersehen: Datenobjekt -> Relation ist nicht die fundamentale Richtung. Die UXP-Lesung lautet: Unterscheidung -> geordnete Relation -> Inschrift -> öffentliche Auslesung. Eine Datei bleibt als benutzbare Datei vollkommen real. Es geht nicht darum, Dateien abzuschaffen. Es geht darum, aufzuhören, das Datei-Objekt als primitive Quelle der Unterscheidungen zu behandeln, durch die die Datei lesbar ist. Diese Differenz wird entscheidend, wenn Code geschrieben wird.

2 Differenz vor Daten

Die kleinste von UXP benötigte Notation ist

|.

Dieses Zeichen bezeichnet anfangs keinen Wert.

Es markiert Unterscheidung. Wenn die Unterscheidung mit expliziten Polen gelesen wird, schreiben wir A|B.

A und B sind nicht zwei unabhängig fertige Dinge, zwischen die nachträglich ein Separator eingefügt wird. Sie sind relationale Positionen, die durch die Unterscheidung lesbar gemacht werden. Die dritte Ordnung ist die Selbstreferenz der Unterscheidung:

$$|(|)$$

Der Tractatus Principio definiert

$$\Delta := |(|)$$

Delta ist daher kein viertes Objekt, das zu den ersten drei hinzugefügt wird. Es ist die operative Selbstreferenz derselben Unterscheidung. Dieser Punkt ist wichtig genug, um ihn negativ zu formulieren. UXP nimmt nicht an: dann A|B, dann |(|) als drei zeitliche Konstruktionsstadien. Die Darstellung ist sequenziell, weil Sprache sequenziell ist. Die Ordnungen selbst wirken gleichzeitig. Was von einem Softwareschritt zum nächsten wechselt, ist die sichtbare Lesung der Relation, nicht die ontologische Geburt einer neuen Schicht. Warum sollte das einen Programmierer interessieren? Weil Softwarearchitektur uns ständig dazu verführt, Leserollen zu verdinglichen. Eine Klasse heißt `Mark`, eine andere `State`, eine andere `Address`, eine andere `Decoder`, eine andere `Path`. Diese Namen sind nützliche technische Griffe. Sie werden nur dann zu einem begrifflichen Fehler, wenn die Implementierung beginnt, jeden Griff als neue Informationssubstanz mit einem unabhängigen Gesetz zu behandeln. Die UXP-Bedingung ist strenger: unterschiedliche Code-Oberfläche != unterschiedlicher relationaler Ursprung. Dies ist auch der Grund, warum Zahl im UXP-Kernel nicht primär sein kann. Ein Byte-Wert, eine Dezimalziffer oder ein hexadezimaler Glyph ist bereits ein unterscheidbares öffentliches Zeichen. Das Programm mag mit Ganzzahlen operieren, weil Maschinen darin gut sind, aber die Ganzzahldarstellung ist Träger

einer Lesung, nicht das Fundament der Differenz. Im Tractatus Principio ist Zahl eine stabilisierte lesbare Ordnung der Unterscheidung. Band VIII schreibt dies operativ als: n := read Σ_n.

Die sichtbare Zahl ist ein Auslesen. Sie ist nicht das primitive Ding, aus dem die Relation später konstruiert wird.

2.1 Zahl als Takt-Auslesen der Selbstrelation Band I gibt die numerische Bewegung, bevor er uns ein Ziffernsystem gibt. Er schreibt:

$$0 \to 1 \to 1+1 \to 1+1+1 \to 1+1+1+1$$

Die Erklärungsrichtung ist entscheidend.

Die 1 ist bereits ein lesbarer Pol von:

0|1.

Dann bezieht sich die bestehende Relation auf sich selbst.

Band I beschreibt die 1 als beginnend, sich auf sich selbst zu beziehen und sich in der 2 zu spiegeln:

1|1.

Der Fehler wäre, die zwei sichtbaren 1-Zeichen in zwei unabhängig existierende Einheitenobjekte zu verwandeln.

Band IV und VIII geben die strengere Lesung.

Die aktive Selbstreferenz ist:

$$\Delta := |(|)$$

und Delta bildet keine Klassen seiner selbst.

Delta erscheint als Takt.

Sei:

Tj (Δ) eine konkrete geordnete Übergangslesung dieser selben Selbstreferenz bezeichnen. Der Takt-Index ist nur die sichtbare Position des Übergangs. Der Takt selbst liegt im operativen Übergang von einer relationalen Position zur nächsten.

Daher:

$$\Delta[n] := T1\ (\Delta)|T2\ (\Delta)|\ \cdots\ |Tn\ (\Delta)$$

(2a)

Für jeden nichtleeren endlichen Takt-Lauf gilt:

$$\Delta[n] \equiv_\Delta \Delta$$

(2b)

Das Äquivalenzzeichen ist wesentlich.

Es sagt nicht, dass eine Ein-Takt-Spur und eine Vier-Takt-Spur dieselbe Spur sind. Es sagt, dass der vierfache Rhythmus keine vier ontologisch verschiedenen Delta-Objekte erzeugt hat. Der operative Ursprung jedes Übergangs ist dieselbe Selbstreferenz der Differenz. Zahl ist die stabilisierte öffentliche Lesung dieses Rhythmus:

n := *read*num (Δ[n]) .

Die 2 ist daher die öffentliche Umschreibung der Eigenrelation von 1:

$$\mathbf{2} = \mathit{readnum}(T\mathbf{1}(\Delta)|T\mathbf{2}(\Delta))$$

Dieselbe Lesung erklärt, warum Band I die 1 viermal innerhalb der 4 bewahren kann:

1|1|1|1

UXP liest dies als vier unterscheidbare Takt-Positionen desselben selbstrelationalen Ursprungs:

$$4 = readnum(T1(\Delta)|T2(\Delta)|T3(\Delta)|T4(\Delta))$$

Der Ausdruck ersetzt die geometrischen Lesarten von Band I nicht.

(2c)

Die 2 kann weiterhin als Spiegel und Durchmesser gelesen werden, die 3 als dreieckige Stabilisierung, die 4 als viereckige und kombinatorische Relation. Das sind verschiedene Lesarten derselben numerischen Bewegung.

Die entscheidende Typunterscheidung lautet: Takt-Vielfalt ≠ neue Delta-Klasse ≠ Adresstiefe.

(2d)

Die Zahl sagt uns, wie derselbe unterscheidende Rhythmus an einer geordneten Position öffentlich lesbar geworden ist.

Sie sagt uns für sich allein nicht, dass die Lesung in eine andere Referenzschale gewechselt ist.

2.2 Ein Ziffernsystem synchronisiert den Takt auf einer Schale Sobald die numerischen Umschreibungen sichtbare Symbole stabilisiert haben, werden diese Symbole zusammen auf einer Ebene gelesen.

Sei:

$T_{r,0}$, $T_{r,1}$, … , $T_{r,9}$ die zehn lesbaren Takt-Positionen der Dezimalschale auf Referenzebene r sein. Dann: (r)

$\mathcal{N}10 \coloneqq read\text{num}\,(T_{r,0} \,|T_{r,1} \,|\, \cdots \,|T_{r,9}\,)$.

Ihre öffentliche Form ist: (r)

$$\mathcal{N}10 = 0|1|2|3|4|5|6|7|8|9$$

(2e)

Dies ist der entscheidende Schritt, durch den ein Ziffernsystem lesbar wird. Die Symbole sind keine zehn primitiven numerischen Substanzen.

Sie sind zehn stabilisierte öffentliche Positionen desselben Differenz-Rhythmus auf einer gemeinsamen Schale. Der Punkt lässt sich sorgfältig formulieren: Ziffernsystem = gemeinsame öffentliche Leseebene eines geordneten Δ-Takts.

(2f)

Diese gemeinsame Ebene löscht die reicheren relationalen und geometrischen Lesarten der einzelnen numerischen Formen nicht aus. Sie synchronisiert sie.

Deshalb können die Spiegel-Lesung der 2, die dreieckige Lesung der 3 und die vierfache Lesung der 4 alle nebeneinander erscheinen als:

2|3|4 ohne zu implizieren, dass zuerst drei fertige Zahl-Objekte existierten.

Ihre qualitativen Relationen bleiben durch die Spur lesbar.

Ihre Ziffern-Glyphen sind auf derselben öffentlichen Takt-Ebene gelandet. Dies klärt auch die Rolle der Bruchkante:

$9 := \kappa$.

Die 9 schließt die sichtbare dezimale Takt-Schale.

Sie wird nicht zu einem Zähler relationaler Tiefe.

Die nächste öffentliche Form:

10 := 1|0 ist kein elftes Grundzeichen auf derselben Ebene.

Sie ist der sichtbare Hinweis, dass die vollendete Ordnung selbst zu einer neuen Referenz geworden ist. Sei:

$$Sr := Tr,0 \mid Tr,1 \mid \cdots \mid Tr,9$$

die vollendete Schalenrelation bezeichnen.

Ein Schalenübergang tritt ein, wenn diese bereits selbstunterscheidende Ordnung selbst unter eine andere Unterscheidung gestellt wird: Δ

$$\rightarrow Sr{+}1 \ . \ Sr -$$

(2g)

Tiefe gehört daher zur Relation zwischen rekursiv referenzierten Schalenordnungen, nicht zur Tick-Zahl innerhalb einer Schale:

readd Σ = Lesung rekursiver Ordnung-von-Ordnung in der Spur.

(2h)

Genau deshalb ist weder:

depth = #(9) noch:

depth = Parsebaum-Verschachtelung der Zahlennotation gültig.

Das erste verwechselt eine öffentliche Bruchkante mit Referenztiefe.

Das zweite verwechselt eine gewählte Zerlegung mit der Selbstreferenz der ausgeführten Relation. Der Takt und die Tiefe sind verschiedene Lesungen derselben Bewegung: • Takt liest die geordnete Fortsetzung von Δ innerhalb der aktuellen Relation. • Schale liest die stabilisierte gemeinsame Ebene dieser Takt-Positionen. • Tiefe liest die rekursive Selbstreferenz, in der eine bereits differenzierte Schale wieder Referenz wird.

2.3 Konsequenz für UXP UXP muss diese Richtung im Code bewahren.

Der Kernel darf nicht beginnen mit: Digit("2") -> eine interne Relation erfinden, die die Ziffer erklärt

noch mit: R2 = ein Spezialbaum mit besonderer Tiefe

Der strengere ausführbare Typ ist: Unterscheidung -> Delta-Selbstreferenz -> geordnetes Takt-Ereignis -> Spur -> Zahlenauslesung

Schematisch:

$$| \to \Delta \to Tj\ (\Delta) \to \Sigma \to read\mathrm{num}$$

Eine sichtbare Ziffer ist eine Schnittstellen-Auslesung einer geordneten Takt-Position. Ihr öffentliches Symbol kann an einer anderen Notation, Geometrie oder Adressprojektion teilnehmen, aber das Symbol wird nicht zum Kernel-Träger.

Ebenso ist read_d keine Funktion des Glyphen.

Es ist eine Lesung der rekursiven Referenzstruktur, die in der Spur bewahrt wird. Dasselbe Prinzip erreicht nun die Datei.

3 Auf der digitalen Oberfläche ist eine Datei eine numerische Relation

Die digitale Qualifikation ist wichtig.

UXP nimmt nicht ein beliebiges pränumerisches Objekt und wandelt es dann in Zahlen um. Auf einer digitalen Maschinenoberfläche ist die Inschrift bereits diskret geordnet und numerisch lesbar. Für eine gewöhnliche byte-adressierte Datei:

$$IF = b0\ |b1\ |\ \cdots\ |bn{-}1$$

mit

$bk \in \{0, \ldots, 255\}$.

Dies ist bereits eine endliche numerische Relation.

Die Byte-Werte werden dadurch nicht zu fundamentalen Zahl-Objekten erhoben. Sie sind die öffentliche Maschinenlesung unterscheidbarer digitaler Positionen. Die Tractatus-Ordnung bleibt:

| → Δ → Takt → numerische Relation → digitale Inschrift

Eine digitale Darstellung, die sich nicht sichtbar als Byte-Zahlenliste präsentiert, muss dennoch durch eine feste Isomorphie des Lesens auf eine numerische Inschrift bezogen sein:

Fdigital $\cong IN$.

(3a)

Hier bedeutet ≅ keine freie Substitution von Symbolen. Es bedeutet Bewahrung derselben relationalen Ordnung unter einer anderen digitalen Lesung. Band VIII gibt das exakte Kriterium für eine solche Übersetzung: Position der Pole, Bindungsmodus, Ordnung, Öffnungen, Gates, Spur, Adresse und Wiederherstellbarkeit müssen bewahrt werden. Daher muss eine gültige digitale numerische Isomorphie in der entsprechenden Notation erfüllen:

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

(3b)

Für die gewöhnliche Byte-Oberfläche ist die einfachste Lesung bereits numerisch:

$F\text{bytes} \cong b0\ |b1\ |\ \cdots\ |bn-1\ .$

Eine dezimale, hexadezimale oder binäre Darstellung ist eine andere öffentliche Lesung dieser numerischen Relation. Sie ist keine andere Quellontologie. Diese Unterscheidung ist für den Archiver wichtig.

Die dezimale Takt-Schale:

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

ist eine konkrete Ziffern-Ausleseoberfläche, die vom Referenzprofil verwendet wird. Sie darf nicht verwechselt werden mit dem allgemeinen Typ digitaler Eingabe.

Daher:

digitale Darstellbarkeit als numerische Relation ⫧Zulassung zu einem bestimmten UXP-Profil DP . (3c) Jede Datei, die der digitalen Maschine präsentiert wird, ist auf dieser Oberfläche numerisch lesbar. Aber das aktuelle ausführbare Profil ist exakt nur für jene numerischen Relationen, für die seine feste Relation, Gates, Lock und Normalformregel einen gegen-lesbaren Pfad bestimmen.

Das ist die präzise Abdeckungsfrage.

Die vertikalen Striche in:

$$b0\ |b1\ |\ \cdots\ |bn-1$$

bedeuten nicht, dass literale Separator-Bytes in die Datei eingefügt werden. Sie legen die Ordnung unterscheidbarer Positionen frei.

Ordnung zählt.

Grenze zählt.

Und zwei digitale Inschriften, die dieselben numerischen Zeichen in verschiedener Ordnung enthalten, sind verschiedene Inschriften:

$$41|42 \neq 42|41$$

Der UXP Archiver liest die digitale numerische Inschrift daher als Pfadrelation:

$IN \to \gamma\, I\, N \to \Sigma\, I\, N$.

Der Pfad ist keine Metadaten, die an ein realeres Datenobjekt angehängt sind. Die geordneten Unterscheidungen sind der Pfad.

Das ist die erste ausführbare Bedeutung von: Die Regel ist der Pfad.

4 Von der Inschrift zur Pfadrelation

Eine Dateiinschrift wird nicht ausgeführt, indem zuerst jedem sichtbaren Zeichen eine willkürliche reversible arithmetische Transformation zugewiesen wird.

Die geordnete Relation ist bereits ausreichend.

Sei:

$$I = m0 \mid m1 \mid \cdots \mid mn-1$$

Zwischen aufeinanderfolgenden adressierten Positionen liegt eine Kante:

$\varepsilon j \coloneqq Aj-1 \mid Aj$.

Ihre konkrete geordnete Ausführung ist der Takt:

$Tj \coloneqq Step \circ Oj$.

Band VIII ist ausdrücklich darin, dass der Takt nicht der Ganzzahlindex j ist. Der Index markiert nur den geordneten Ort.

Der Takt selbst liegt im Übergang: Tj

$Aj-1 \longrightarrow Aj$.

Die Inschrift bestimmt daher einen Pfad:

$\gamma I = (\varepsilon 1\, , \varepsilon 2\, , \ldots\, , \varepsilon n-1\,)$, und der Pfad bestimmt eine Spur:

$\Sigma j+1 = \Sigma j \oplus Hj$.

Der lokale Beitrag ist:

$Hj = \chi \mathrm{vis}\,(\varepsilon j\,)\chi \mathrm{gate}\,(\varepsilon j\,)w(\varepsilon j\,)$.

(1)

Dies ist das ausführbare Pfadgesetz.

Nichts daran erfordert eine affine Permutation, eine inverse Kodierfamilie oder einen verborgenen numerischen Seed.

Die Regel ist der Pfad, weil die geordneten Übergänge die Relation sind, die ausgeführt wird.

5 Was Pfadkomposition tatsächlich bewahrt

Wenn eine Inschrift zur Ausführung geteilt wird:

$I = U \mid V$, erzeugt der Split keine zwei unabhängigen Informationsobjekte.

Er erzeugt zwei aufeinanderfolgende Pfadregionen:

$\gamma I = \gamma U \parallel \gamma V$, wobei $\parallel$ ordnungserhaltende Fortsetzung bezeichnet.

Ihre Spurrelation ist entsprechend:

$\Sigma(\gamma I) = \Sigma(\gamma U \parallel \gamma V)$.

(2)

Im Allgemeinen gilt:

$\gamma U \parallel \gamma V \neq \gamma V \parallel \gamma U$.

Die Bedeutung liegt nicht darin, dass Funktionskomposition eine externe Algebra ist, die auf die Datei gesetzt wird. Sie liegt darin, dass die Sequenz der Takte selbst bedeutungstragend ist. Die Spur erinnert den geordneten Vollzug, weil:

$\Sigma j{+}1 = \Sigma j \oplus Hj$ auf die bereits etablierte Spur wirkt.

Deshalb kann technisches Stückeln die Ausführungsgruppierung ändern, ohne mathematische Archivtiefen zu werden. Eine Arbeitergrenze bedeutet: diese zusammenhängende Pfadregion getrennt ausführen.

Sie bedeutet nicht: hier beginnt eine neue UXP-Entität.

6 Technische Schnitte sind keine mathematischen Schnitte

Eine konventionelle Implementierung lässt ihr technisches Layout oft in ihre mathematische Erzählung durchsickern. Eine Blockgröße wird gewählt. Bald wird der Block zu einer „Einheit“. Eine Cache-Grenze wird zu einer „Ebene“. Eine Dateipartition wird zu einem ontologischen Teil der Daten. UXP muss dieser Drift widerstehen. Angenommen

```
I = I1|I2|…|Ik.
```

Wiederholte Anwendung des Kompositionstheorems ergibt GI = GI k ◦ · · · ◦ GI 2 ◦ GI 1 . (2) Nichts in (2) spezifiziert die physische Größe der Stücke. Daher kann der Archiver aus praktischen Gründen eine technische Stückgröße wählen: Prozessaufwand, Cache-Verhalten, Speicherdruck, Warteschlangenlatenz, Arbeiterauslastung. Diese Entscheidungen können die Laufzeit ändern.

Sie ändern nicht die geordnete Relation, die gelesen wird. Diese Unterscheidung macht Streaming möglich, ohne den Streaming-Vollzug in einen neuen mathematischen Mechanismus zu verwandeln. Eine Puffergrenze sagt: hier für die Ausführung getrennt lesen.

Sie sagt nicht: a new UXP object begins here.

Die Software darf schneiden. Die Relation muss es nicht.

7 Assoziativität öffnet den Arbeiterbaum

Funktionskomposition ist assoziativ: $G_W o G_V$ o G_U = G_W o $G_V o G_U$. (3) Assoziativität erlaubt keine willkürliche Umordnung.

Sie erlaubt Umgruppierung, während die Ordnung bewahrt bleibt.

Diese Differenz ist das Fundament der Arbeiterarchitektur. Für vier aufeinanderfolgende Relationen, G1, G2, G3, G4, kann das Programm sie auf mehrere ordnungserhaltende Weisen assoziieren. Es kann eine Linksfaltung, einen ausbalancierten Baum oder eine laufende Kaskade verwenden. Das Ergebnis bleibt dieselbe vollständige geordnete Relation. Der Archiver verwendet eine vierstellige laufende Kaskade. Vier aufeinanderfolgende Relationen auf einer Ebene binden zu einer Relation auf der nächsten Ebene:

$$G1|G2|G3|G4 -> G4 o G3 o G2 o G1$$

Vier vollendete Relationen dieser nächsten Ebene können sich sofort erneut binden. Das wichtige Wort ist sofort. Die gesamte untere Ebene muss nicht abgeschlossen sein, bevor eine höhere Ebene beginnen kann. Sobald vier aufeinanderfolgende Relationen existieren, kann die nächste Ebene wirken. Die sichtbare Ausführung ist sukzessiv. Die Relation ist rekursiv über Ebenen hinweg aktiv. Deshalb ist „Kaskade“ mehr als ein Leistungsetikett. Sie ist eine Implementierungsform, die nahe an der Tractatus-Lesung rekursiver Selbstreferenz bleibt: Spätere relationale Organisation muss nicht warten, bis eine vollständig materialisierte niedrigere Welt zuerst als Objekt existiert.

8 Die Vier-Kaskade

Die Vier-Kaskade lässt sich am einfachsten als laufender Strom verstehen. Eingabefragmente treffen ein: I1 I2 I3 I4 I5 I6 I7 I8 … Ebene 0 liest jedes Fragment in eine geordnete Pfadrelation: G1 G2 G3 G4 G5 G6 G7 G8 … Nachdem G1 bis G4 existieren, kann Ebene 1 bereits bilden H1 = G4 o G3 o G2 o G1.

Nachdem G5 bis G8 existieren, bildet sie H2 = G8 o G7 o G6 o G5.

Der untere Strom läuft weiter.

Sobald vier aufeinanderfolgende H-Relationen existieren, bindet die nächste Ebene sie erneut. Kein zusätzlicher „höherer Kodierer“ wird eingeführt. Dieselbe Kompositionsregel kehrt wieder. Dies gibt der Architektur eine einfache Form: endliche Inschrift | v technische Stromschnitte | v lokale Pfadrelationen | v geordnete Vier-Kaskade | v vollständige Relation | v öffentlicher Anwendungsträger.

Die Gegenrichtung spiegelt dieselbe Struktur in Gegenorientierung.

Dort beginnt das übliche Encoder/Dekoder-Bild zu zerbrechen.

9 Umkehrung ist keine zweite Ontologie

Konventionelle Software-Sprache benennt die beiden Richtungen oft als „encode“ und „decode“. UXP kann zwei Benutzerhandlungen sichtbar machen - readressieren und gegenreadressieren -, weil ein Benutzer beide Orientierungen anfordern kann. Diese Schnittstellenbefehle dürfen nicht zu zwei mathematischen Mechanismen werden. Mathematisch ist das Ziel jedoch strenger. Die Umkehrung darf nicht heimlich zu einer zweiten Informationsroute werden, die Rekonstruktionswissen trägt, das die Vorwärtsrelation nicht enthielt. Für komponierbare invertierbare Relationen gilt:

$$(G_V o G_U)^(- 1) = G_U^(- 1) o G_V^(- 1). (4)$$

Gleichung (4) ist das Spiegelgesetz. Der Vorwärtspfad U dann V wird in umgekehrter Reihenfolge gegengelesen. Das ist keine ungewöhnliche Algebra. Die Bedeutung von UXP liegt in dem, was verweigert wird: Es besteht keine Notwendigkeit, einen zweiten Pfad als Erklärung der Wiederherstellung zu erfinden. Dieselbe etablierte Relation wird in der anderen Orientierung gelesen. Der Benutzer kann diese Operation „wiederherstellen“ nennen. Die mathematische Lesart ist: Gegenreadressierung.

Diese Unterscheidung wird völlig explizit, sobald die Relation selbstorientierend gemacht wird.

10 Warum die erste beschränkte Inverse nicht genügte

An diesem Punkt kann ein Leser die entscheidende praktische Frage stellen: Wenn nur ein endgültiges öffentliches Auslesen bleibt, woher kann das Gegenlesen wissen, welche Fortsetzung zulässig ist?

Die falsche Antwort wäre: weil die endgültige Zahl irgendwie ihren historischen Pfad enthält. UXP erhebt keinen solchen Anspruch.

Die Relation muss stattdessen die Orientierung beschränken, die für ihr eigenes Gegenlesen erforderlich ist. Angenommen, ein Relationsraum S hat zehn lokale Aktionen

$T0 , … , T 9 : S → S$ und jede Aktion ist eine Bijektion des ganzen Raums. Dann gilt

$Tj (S) = S$ für jedes j. Für jeden Endpunkt y kann jede Orientierung einen Vorgänger besitzen. Algebraische Invertierbarkeit allein identifiziert daher nicht, welcher Vorgänger zum tatsächlichen Pfad gehört. Die UXP-Anforderung ist stärker: die fixe Perspektive muss die zulässige Fortsetzung einschränken.

Band VI formuliert die Mehrdeutigkeit als Kandidatenrelation

$\delta(x; Γ) = \{y ∈ Adr \mid N ext(x, y) ∧ Allow Γ (x, y)\}$.

Bedingungskomposition ist konjunktiv:

$Allow Γ1 ⊗ Γ2 = Allow Γ1 ∧ Allow Γ2 .$

Daher gilt

$$\delta(x;\ \Gamma 1 \otimes \Gamma 2\ ) \subseteq \delta(x;\ \Gamma 1\ )$$

(5)

Unter Fixierung φ wählt der kanonische Selektor $Sel\varphi$ die stabile Fortsetzung aus der verfeinerten Kandidatenmenge. Determinismus ist damit keine spätere Optimierung über vollendeten Objekten; er ist der Abschluss von Mehrdeutigkeit unter einer fixen Referenz (Tractatus Principio, Band VI, §§15-16). Die entscheidende Frage lautet daher nicht: Ist jede lokale Abbildung invertierbar?

Sondern: Gibt es unter dem fixen Profil genau eine zulässige gegenorientierte Fortsetzung? Diese Frage beherrscht jeden späteren UXP-Gültigkeitsanspruch.

11 Eine selbstorientierende Takt-Relation

Das kleinste lokale UXP-Beispiel sollte mit dem Takt der Differenz beginnen, nicht mit zehn fertigen Zahlzeichenobjekten.

Der operative Ursprung ist:

$$\Delta\ :=\ |(|)$$

Die dezimale Zahlzeichenhülle legt zehn öffentliche Takt-Positionen frei:

$T0$, $T1$, … , $T9$.

Jedes Tj ist eine konkrete geordnete Lesung desselben Δ:

$Tj \equiv \Delta\ \Delta$.

Das sichtbare Zahlzeichen ist:

j = *read*num (Tj).

Fixiere eine Takt-Position Tm .

Die anderen neun Positionen derselben Hülle sind ihre unmittelbaren Gegenpositionen:

$$\boldsymbol{CT(Tm) = Tm|Tn \mid n \neq m}$$

Daher:

$$|CT\ \ (Tm\ \ )|\ =\ 9$$

Die neun Terme sind Relationen zwischen Takt-Positionen, nicht neun alternative Integer-Nutzlasten. Ihre Orientierung folgt demselben numerischen Rhythmus, der die öffentliche Ebene lesbar macht:

$$T0\ \rightarrow\ T\ 1\ \rightarrow\ \cdots\ \rightarrow\ T\ 9\ \rightarrow\ T\ 0$$

Diese zyklische Notation definiert Delta nicht durch Arithmetik modulo zehn. Sie drückt den öffentlichen Abschluss der dezimalen Takt-Ebene aus.

Unter einem fixen Profil P bleibt die zulässige Fortsetzung:

$\delta P\,(x) = \{y \mid N\,extP\,(x, y) \wedge AllowΓP\,(x, y)\}$, und die Fixierung φ schließt verbleibende Mehrdeutigkeit durch den kanonischen Selektor. Die ausgewählte Fortsetzung erzeugt keine neue Art von Delta.

Sie ist ein anderer geordneter Takt derselben Selbstreferenz.

Dies ist die kleinste ausführbare Bedeutung von: Die Regel ist der Pfad.

12 Vom ganzen Pfad zur öffentlichen Unter-Ziel-Relation

Die Kaskade beginnt nicht damit, eine lange numerische Inschrift in kleinere numerische Entitäten zu schneiden.

Sie beginnt mit der ganzen Inschrift:

$$I(r) = m0|m1|\cdots|mNr - 1$$

Die ganze Inschrift wird als ein geordneter Pfad gelesen:

$I\,(r) \longrightarrow \gamma I\,(r)$.

Die Adresse ist der Ort, der durch diesen ganzen Pfad unter dem fixen Profil erreicht wird:

$AdrP\,(I\,(r)) \coloneqq locateP\,(\gamma I\,(r))$.

(8)

Die nächste öffentliche numerische Relation ist eine Lesung dieser Adresse:

$I\,(r{+}1) \coloneqq readP\,(AdrP\,(I\,(r)))$.

(8a)

ganze Inschrift ≠ Pfad ≠ Adresse ≠ numerisches Auslesen.

(8b)

Also:

Die kürzere Relation ist keine Sammlung von Adressen von Stücken der früheren Relation. Sie ist die frühere Relation, gelesen an einer anderen Adresse.

Band IX nennt dieselbe Bewegung als: Rk

$IN \dashrightarrow IN{-}k$,

$IN{-}k \equiv_{\Sigma} IN$.

12.1 Es gibt keine fixe Zielrelation Sei:

$U := 10$.

Die einzige öffentliche Zielbedingung ist:

$StopU\ (I) \Longleftrightarrow width(I) \leq U$.

(8c)

Wenn die aktuelle Relation breiter als zehn Positionen ist, readaddressiert dieselbe fixe Relation die ganze aktuelle Inschrift.

Keine gewählte zehnstellige Zeichenkette ist ein Ziel.

Kein Wurzelwert wird im Voraus ausgewählt.

Keine Hüllenanzahl wird geliefert.

Zehn ist eine obere Grenze der Breite der öffentlichen Relation. 12.2 Unter-Ziel ist keine Blockstruktur Ungültig: Quelle in Gruppen aufteilen jede Gruppe adressieren die Gruppen als Kindrelationen mitführen wiederholen

Das würde die lange Inschrift in einen Baum kleinerer Relationsobjekte verwandeln. Band IX weist eine AST-Ontologie ausdrücklich zurück.

Gültig: ganze Relation -> ganzer Pfad -> relationale Adresse -> numerisches Auslesen dieser Adresse

und in Gegenorientierung: numerisches Auslesen -> relationale Adresse -> ganzer Pfad -> ganze Relation

12.3 Fixes Byte-Referenzprofil Die digitale Quelle ist:

$$I = b0\ |b1\ |\ \cdots\ |bn-1$$

$bj \in \{0, \ldots, 255\}$.

Eine fixe Wurzeladresse wird vor der Ausführung gewählt:

$a0 = \odot\varphi$.

Jedes Byte-Zeichen treibt ein und dieselbe Adressrelation um einen Schritt weiter:

$aj+1 = N\ extP\ (aj\ ,\ bj\ )$.

(8d)

Die Adresse der ganzen Inschrift ist:

$AdrP\ (I) = an$.

Version 0.14.1 implementiert ein gewurzeltes 256-Wege-Adressnetz mit einem eindeutigen gegenorientierten Vorgänger für jede Nicht-Wurzel-Adresse. Die konkrete shift/append-Schreibweise in Python und C ist eine technische Notation dieser Next_P- Relation, kein neues

Primitiv des Tractatus.

12.4 Dieselbe Regel bestimmt den Gegenpfad Für jede Nicht-Wurzel-Adresse gilt:

$P\ redP\ (a'\ ) = (a, m)$

$\Longleftrightarrow$

$N\ extP\ (a, m) = a'\ .$

Also:

$$an \to an-1 \to \cdots \to a0 = \odot\varphi$$

Die in Gegenorientierung angetroffenen Zeichen reproduzieren die ursprüngliche geordnete Inschrift. Der Gegenpfad endet an der fixen Wurzelrelation.

Er endet nicht nach einer gelieferten Quelllänge oder einer gelieferten Tiefe. 12.5 Exaktheit auf endlichen Byte-Inschriften Weil Next_P genau eine exakte Pred_P-Gegenrelation besitzt, ergibt Induktion über jede endliche Byte-Inschrift:

$counterP\ (readP\ (AdrP\ (I))) = I.$

(8e)

Dies gilt für:

$$DP = \bigcup \{0, \ldots, 255\}^n \ .\ n \geq 0$$

Die Tests sind Ausführungsprüfungen derselben Relation. 12.6 Relationsbreite ist nicht Notationslänge UNDER_TARGET liest die Anzahl der Positionen in der öffentlichen numerischen Relation.

Es zählt nicht die Zeichen, die von einer textuellen Notation einer numerischen Adresse verwendet werden. Daher: Relationsbreite ≠ Glyphenlänge einer Zahlzeichennotation.

(8f)

Eine numerische Adresse kann dezimal, hexadezimal oder in einer anderen isomorphen Zahlzeichennotation dargestellt werden.

Die Adresse und ihr Zahlzeichen bleiben verschiedene Typen:

$Addr \neq read(Addr)$.

12.7 Die dezimale Takt-Hülle behält ihre Rolle Die zehn öffentlichen Zeichen:

$$0|1| \cdots |9$$

bleiben die gemeinsame Zahlzeichenebene.

Die 9 bleibt die sichtbare Bruchkante:

$9 := \kappa$.

Und:

$10 := 1\,|\,0$ markiert weiterhin einen Wechsel der Referenzrelation statt eines elften primitiven Zeichens. Was sich ändert, ist nur dies: die dezimale Hülle wird nicht länger für die erforderliche Gestalt der Quellrelation gehalten. 12.8 Globales readaddressing Die operative Sequenz ist nun:

$I \to \gamma I \to \Sigma \to AdrP \to N\,F\varphi \to read$.

Band VIII fordert:

$recover\mathbf{read}(\boldsymbol{\Sigma}) \equiv \Sigma\ \Sigma$.

Band IX fordert: Rk

$IN \dashrightarrow IN{-}k$,

$IN{-}k \equiv \Sigma\ IN$.

Die ausführbare Referenz liest beides zusammen:

ganze numerische Relation ⟷ Ganzpfad-Adresse ⟷ öffentliches Unter-Ziel-Auslesen.

13 Die Regel ist der Pfad

Der Satz Die Regel ist der Pfad. kann nun als ausführbarer Vertrag formuliert werden.

Für ein fixes Profil P gilt:

$\delta P\,(x) = \{y \in Adr \mid N\,extP\,(x, y) \wedge Allow\Gamma P\,(x, y)\}$.

(8g)

Die fixe Referenz φ liefert die kanonische Fortsetzung

$xn{+}1 = Sel\varphi\,(xn\,, \Gamma P\,)$.

(9)

Der Pfad wird daher rekursiv erzeugt:

$\gamma P\,(x0\,) = (x0\,, x1\,, x2\,, \ldots)$.

Keine serialisierte Liste der Entscheidungen ist erforderlich, wenn dasselbe fixe Profil sie regeneriert. 13.1 Mehrdeutigkeit wird zu einer weiteren Unterscheidung Wenn

$$|\,\delta P\ \ (x)\,|\ >\ 1$$

ist die aktuelle Relation noch nicht geschlossen.

UXP zwingt die Alternativen nicht in einen Wert. Die ungelöste Unterscheidung wird zum Material einer anderen Referenzrelation: Mehrdeutigkeit auf Tiefe $r \Longrightarrow$ neue Unterscheidung auf Tiefe $r + 1$.

(9a)

Dies ist die Anwendungslesung von

$$\Delta := |(|)$$

Der Ebenenwechsel ist kein vierter Operator, der hinter die Differenz gesetzt wird. Er ist ein operatives Erscheinen der Selbstreferenz der Unterscheidung. 13.2 Gegenlesen ist dieselbe Relation in Gegenorientierung Die vollendete öffentliche Relation ist:

$I \to \gamma I \to \Sigma \to \Pi \to \bowtie \to Lock\ 3 \to N\ F\varphi \to LNF$.

Gegenlesen beginnt bei der öffentlichen Normalform, aber es entpackt kein gespeichertes Paar wie etwa:

(depth, payload).

Stattdessen:

$LNF \rightsquigarrow N\ F\varphi \rightsquigarrow [\Sigma]{\equiv}\Sigma$.

(9b)

Wo die gewählte Normalform Pfad- und Inschriftenordnung bewahrt, ist dieselbe Relation dann in Gegenorientierung lesbar:

$$[\Sigma]{\equiv}\Sigma \ \leftrightarrow\ \gamma I \ \leftrightarrow\ I$$

Für rekursives Hüllen-readaddressing wird dieselbe Aussage auf jede adressierte Relation angewendet:

$$[\Sigma(r{+}1)\ ]{\equiv}\Sigma \ \leftrightarrow\ A(r) \ \leftrightarrow\ [\Sigma(r)\ ]{\equiv}\Sigma$$

Tiefe wird nicht geliefert, um zu wählen, wie oft diese Operation ausgeführt wird. Auf jeder Ebene gilt:

$depth := readd\ \boldsymbol{\Sigma}$.

Die Gegenbewegung liest daher die bereits etablierte Relation, anstatt eine unabhängige Rekonstruktionsroute zu konsultieren.

Der Pfad wird nicht von einer Anwendungsdatenbank „erinnert“. Wo Exaktheit beansprucht wird, bleibt er wiederherstellbar, weil die Pfadordnung durch die gebundene Normalform bewahrt wird. 13.3 Abschluss ist ebenfalls Teil der Regel Ein Pfad ist nicht exakt, wenn der Gegenleser weiß, wie er fortsetzen soll, aber nicht, wann die Quellinschrift geschlossen ist.

Daher: die Regel bestimmt sowohl Fortsetzung als auch Abschluss.

(9c)

Ein gültiges Archivprofil muss seine Terminierung aus der gebundenen Relation selbst ableiten. Quelllänge, Blattanzahl oder ein äquivalenter dateispezifischer Stoppwert dürfen nicht als unabhängiges Wiederherstellungsargument geliefert werden. 13.4 Starke Wiederherstellung Die starke Archivbedingung auf Spurebene lautet

$recoverP\ (L10\ (I)) \equiv\Sigma\ \Sigma(\gamma I\ )$.

(9d)

Die exakte Rekonstruktion der ursprünglichen Inschrift erfordert die stärkere Bedingung, dass Pfadordnung und Inschriftenordnung durch die ausgewählte Normalform repräsentiert bleiben:

$counterP\ (L10\ (I)) = I$.

(9e)

Diese Unterscheidung folgt dem Typvertrag des Tractatus: Wiederherstellbarkeit der Spurenklasse und exakte Einzelpfad-Rekonstruktion sind verwandt, aber nicht automatisch identisch. „Die Regel ist der Pfad“ bedeutet daher: • kein dateispezifisches Pfadobjekt wird transportiert; • kein verstecktes Nachschlagen liefert den Vorgänger; • keine externe Tiefe liefert die Hüllenanzahl; • keine externe Länge liefert den Abschluss; • dasselbe fixe Profil bestimmt die zulässige gegenorientierte Entfaltung.

14 Was ist tatsächlich öffentlich?

Die öffentliche Grenze enthält das vollständige numerische Unter-Ziel-Auslesen. öffentliche Relation: gespeicherter Quellblock: gespeicherte Pfadliste: gespeicherte Arbeiter-/Stück-Karte: gespeicherte zweite Gegenroute: gespeicherter Tiefen-Sidecar: gespeicherter Quelllängen-Sidecar: gespeicherte persistente Lage-Datei: gemeinsame Anforderung:

vollständiges read$\mathbf{Addr}_{P}$ keine keine keine keine keine keine keine fixes UXP-Profil P

Die Typtrennung lautet: öffentliches numerisches Auslesen ≠ Adresse ≠ Spur ≠ Quellcontainer.

(10a)

Version 0.14.1 testet diese öffentliche Relation direkt.

Die Dateisystem-Schreibweise des Auslesens ist eine technische Projektion und nicht identisch mit mathematischer Relationsbreite.

15 Gemeinsame Relation ist keine dateispezifische Fracht

Ein skeptischer Leser sollte dennoch fragen: Was braucht die Anwendung außer der öffentlichen Wurzel?

Sie braucht das fixe UXP-Leseprofil, unter dem diese Wurzel sinnvoll ist. Das Profil ist gemeinsames ausführbares Gesetz, nicht erinnerte Quellhistorie. Schematisch fixiert es: • zulässige direkte Fortsetzungen $N\ ext$; • das Constraint-Feld Γ und seine Joins; • den Referenzschnitt φ; • Projektionskanäle Πi ; • die Bindungsregel ⋈; • den $Lock$ 3-Abschluss;

• die Normalformregel $N F\varphi$; • Hüllenwechsel- und Tiefenleseregeln; • die äußere Tiefenverschachtelungsregel; • die zulässige Archivdomäne D; • die Abschlussbedingung, die vom Gegenlesen verwendet wird.

Dies sind Regeln des Lesens.

Dementsprechend gilt: gemeinsame Anwendungsrelation ≠ persistierte relationsbezogene Historie pro Quelle.

(11)

Eine .uxp_lage-Datei, eine nach Wurzeln indizierte Datenbank, ein gespeicherter Eltern-Kind-Graph, ein Quellhash-Nachschlagewerk oder eine versteckte Abbildung von öffentlicher Wurzel auf früheren Vorwärtslauf wäre ein anderer Anwendungstyp.

Der öffentliche Vertrag lautet daher: fixes Anwendungsprofil öffentliches L_NF .uxp body dateispezifische Quellnutzlast dateispezifisches Pfadlog dateispezifischer Tiefenparameter dateispezifischer Längenparameter persistente Vorwärtshistorie

= gemeinsames ausführbares Gesetz = dateispezifisches öffentliches Auslesen = leer = keine = keine = keine = keine = keine

Gegenlesen in einem frischen Prozess ist der entscheidende Test.

Nachdem eine Orientierung der Relation geendet hat, muss ein neuer Prozess, der dasselbe fixe Programm/Profil und nur das öffentliche .uxp-Artefakt enthält, in der Lage sein:

- 1. die vollständige öffentliche Normalform zu lesen;

- 2. sie zur entsprechenden Spurenklasse gegenzulesen;

- 3. Tiefe, Adresse und Orientierung aus dieser Spur zu lesen statt aus gespeicherten Feldern;

- 4. derselben zulässigen Relation in Gegenorientierung zu folgen;

- 5. Abschluss aus der Relation selbst zu erkennen;

- 6. die Quellinschrift zu rekonstruieren, wo das Profil Exaktheit beansprucht.

Wenn eine fehlende Unterscheidung durch zuvor befüllten Anwendungszustand geliefert wird, hat die Implementierung regelgeneriertes Gegenlesen durch Speicherung ersetzt. Der ausführbare Vertrag lautet daher nicht: forward run -> retained execution trace -> reconstruction run

sondern:

RP (s, READADDRESS) = s'

$\Longleftrightarrow$

RP (s' , COUNTERREAD) = s.

(11a)

READADDRESS und COUNTERREAD benennen an der Schnittstelle angeforderte Orientierungen.

Sie benennen nicht zwei mathematische Mechanismen.

Wo *P* die Relation auf ihrer zulässigen Domäne injektiv macht, ist die Regel selbst der Pfad.

16 Die Artefaktgrenze folgt der öffentlichen Relation

Frühere Builds verwendeten einen zehnstelligen Dateinamen als bequeme öffentliche Oberfläche. Das ist nicht länger die mathematische Zielbedingung.

Version 0.14.1 macht die numerische Unter-Ziel-Relation durch einen Streaming-Ausführungspfad sichtbar und verifiziert Gegenlesen aus derselben öffentlichen Relation. Die konzeptuelle Artefaktgrenze bleibt: öffentliches Normalform-Auslesen dateispezifische Quellnutzlast: keine dateispezifisches Pfadlog: keine dateispezifische Tiefe: keine dateispezifische Quelllänge: keine

Eine entscheidende Unterscheidung wird jetzt explizit: [öffentliche Relation ≠ lineare Dateisystem-Schreibweise dieser Relation.]

Die aktuelle Demonstration projiziert weiterhin eine adressierte numerische Position in eine vollständige hexadezimale Zeichenkette.

Diese hexadezimale Zeichenkette ist eine Schnittstellenprojektion.

Ihre Glyphenlänge ist nicht die Breite der öffentlichen Relation, und sie ist nicht der Vollzug, durch den die Relation readaddressiert wird.

Die Schnittstelle kann sich daher ändern, ohne die Pfadrelation zu ändern, vorausgesetzt, dass die neue öffentliche Lesung für dieselbe Spurenklasse wiederherstellbar bleibt.

Eine Dateisystembeschränkung darf niemals durch versteckten Rekonstruktionszustand repariert werden.

17 Der UXP-Archiver als Anwendung

Die sichtbare Vorwärtsbewegung ist: digitale Quelle -> geordnete numerische Inschrift -> ganze Inschrift als ein Pfad -> relationale Ganzpfad-Adresse unter fixem P -> numerisches Auslesen dieser Adresse -> Stopp, wenn Relationsbreite <= 10

Die Gegenorientierung ist: öffentliche numerische Relation -> dieselbe relationale Adresse -> eindeutiger Vorgängerpfad -> ursprüngliche numerische Inschrift -> ursprüngliche digitale Datei

Es erscheint kein verschachtelter Relationsbaum.

18 Was geschieht, wenn ein Nutzer eine Datei readadressiert

Für:

$$I(F) = b0|b1|\cdots|bn-1$$

wenn:

$n \leq 10$, liegt die Inschrift bereits unter dem Ziel.

Wenn:

$n > 10$, wird die ganze Inschrift als ein Pfad gelesen:

$I(F) \rightarrow \gamma F$.

Unter dem festen Profil:

$a0 = \odot \varphi$,

$aj{+}1 = N\ extP\ (aj\ ,\ bj\ )$.

Der erreichte Ort ist:

$AdrP\ (F\ ) = an$.

Seine öffentliche numerische Auslesung ist:

$LF := read(AdrP\ (F\ ))$.

Für das aktuelle Byte-Profil:

$width(LF\ ) = 1 \leq 10$ für jede Quelle, die breiter ist als die Unter-Ziel-Grenze.

Das Gegenlesen folgt Pred_P, bis die feste Wurzelrelation erreicht ist, und rekonstruiert:

$$b0\ |b1\ |\ \cdots\ |bn{-}1$$

Somit:

$counterP\ (LF\ ) = I(F\ )$.

(12)

Es wird keine Quelllänge angegeben.

19 Streaming ist der Vollzug des ganzen Pfades

Streaming-Vollzug ist keine spätere Optimierung, die UXP hinzugefügt wird. Es ist in der Takt-Struktur des Pfades bereits implizit.

Für die digitale Inschrift:

$$I = b_0|b_1|\ldots|b_n$$

wird die aktuelle Adresse fortgesetzt durch: [a_{j+1}=Next_Pa_j, b_j.] Der nächste Takt benötigt daher die bereits erreichte Relation und das nächste Zeichen. Er benötigt nicht, dass die ganze zukünftige Datei zuerst materialisiert wird. Dies ist genau der Unterschied zwischen einem

Pfad-Vollzug und einem Objektcontainer. Beim Takt j ist a_j bereits die Adresse, die durch den bis zu diesem Punkt vollzogenen ganzen Pfad erreicht wurde. Das nächste Byte erzeugt keine neue Unterrelation: [Streaming-Vollzug ≠ Blockontologie.] Technisches Puffern kann auftreten.

Ein Leser kann Bytes, Betriebssystem-Seiten oder I/O-Chunks empfangen.

Keine dieser technischen Schnitte wird zu einem mathematischen Schnitt, es sei denn, die Relation selbst macht ihn zu einem.

Dasselbe gilt rekursiv für die Kaskade: Eine wirksame Relation kann fortgesetzt werden, sobald ihr nächster Takt bestimmt ist. Ein abgeschlossenes gigantisches Ausführungsobjekt ist keine Voraussetzung für die nächste relationale Fortsetzung.

Dies ist bereits die Logik der Schleife aus Band VIII: für takt, kante in aufzählen(pfad): beitrag = schritt_lesen(kante, takt)

Die Schleife stellt keine unabhängigen Pfadstücke her.

Sie vollzieht die Relation durch geordnete Takte.

Band VI gibt dieselbe Richtung von der Seite der Fortsetzung: Der Pfad wird im Gehen erzeugt, und der nächste Zustand wird aus dem aktuellen relationalen Zustand unter dem festen Profil bestimmt.

Damit lautet die operationale Regel: nächster Takt

[aktuelle Relation − −−−−→ fortgesetzte aktuelle Relation.] Nicht: vollständige Datei -> Riesenobjekt -> relationale Ausführung beginnen

Die Regel ist der Pfad, nicht die Vormaterialisierung des Pfades.

20 Leistung und Verifikation

Version 0.14.1 korrigiert eine Implementierungsdrift im vorherigen C-Benchmark. Version 0.14.0 verwendete: vollständige Datei -> Wurzelbyte voranstellen -> BN_bin2bn$entire_file$ -> BN_bn2hex(...)

Dies war eine Ausführungsabkürzung.

Es war nicht die Architektur.

Ein 64-MiB-Fehlschlag dieses monolithischen BIGNUM-Pfades belegte daher keine Grenze von UXP, keine Grenze des Readaddressings und keinen Bedarf an einer neuen Streaming-Methode. Die Relation war bereits streambar.

Version 0.14.1 führt den produktiven Pfad inkrementell aus und entfernt BN_bin2bn$entire_file$ aus diesem Pfad.

Der korrigierte Benchmark berichtet exakte Roundtrips in frischen Prozessen für: 16 MiB 64 MiB 128 MiB

BESTANDEN BESTANDEN BESTANDEN

Bei 16, 64 und 128 MiB bleibt das gemessene RSS-Speichermaximum in der Benchmark-Umgebung ungefähr bei 4.4 MiB.

Die relevante Schlussfolgerung lautet daher: [Quellgröße ≠ quellgroßes Ausführungsobjekt im RAM.] Das bedeutet nicht, dass jede Schnittstellen-Projektion bereits optimal ist. Die aktuelle Demo schreibt die öffentliche Adressauslesung noch als eine vollständige lineare hexadezimale Zahl.

Das ist eine andere Frage.

Der Benchmark muss daher unterscheiden zwischen: A. Streaming-Vollzug der Relation B. physischer Serialisierung einer öffentlichen Zahlennotation

Ein Fehlschlag oder eine Ineffizienz von B ist nicht automatisch ein Fehlschlag oder eine Ineffizienz von A. Die Verifikation bleibt geordnet als: Typtrennung -> Pfad der ganzen Relation -> taktweise Fortsetzung -> öffentliche Breite unter der Zielgrenze -> Gegenlesen im frischen Prozess -> Python/C-Parität -> Mutationskontrollen -> Skalierung -> Schnittstellen-Projektionsaudit

Der Roundtrip bleibt notwendig, aber nicht hinreichend.

21 Warum der Durchsatz zählt

Durchsatz zählt, weil UXP benutzbar sein soll, nicht bloß kontemplativ.

Aber die Ordnung der Verifikation ist festgelegt: Typkorrektheit -> Gegenlesen im frischen Prozess -> selbstschließende Rekonstruktion -> Domänenexaktheit -> Skalierung -> Durchsatz

Ein Programm, das schnell ist, weil es eine verborgene Wurzeldatenbank konsultiert, hat die UXP-Relation nicht implementiert.

Ein Programm, das den korrekt erzeugten Strom erst rekonstruiert, nachdem ihm die ursprüngliche Quelllänge mitgeteilt wurde, hat die vollständige Archivschließung noch nicht implementiert. Ein Programm, das die Tiefenverschachtelung validiert, aber keine gleiche Quelle-zu-Öffentlich-Relation hat, hat eine Schicht der Architektur validiert, nicht Archivierung beliebiger Dateien. Das Entwicklungsziel ist daher präzise: Kann dieselbe relationszuerst arbeitende Laufzeit die beanspruchte Quelldomäne mit praktischem Dateiwerkzeug-Durchsatz verarbeiten und zugleich jeden dateispezifischen Rekonstruktionszustand außerhalb des öffentlichen Normalform-Labels vermeiden?

Nur Messungen nach dieser Prüfschwelle sind Produkt-Durchsatzmessungen.

22 Warum Kompression eine Sackgasse ist

Wir können nun zum Titel zurückkehren.

Kompression fragt: Wie kann die Darstellung dieses Datenobjekts reduziert werden?

UXP fragt: Wie ist diese Inschrift als diese Inschrift lesbar, und wie kann diese Lesung readadressiert werden?

Die beiden Fragen können zu Programmen führen, die nebeneinander auf derselben Maschine liegen. Sie beginnen nicht aus derselben erklärenden Richtung.

Kompression setzt ein Informationsobjekt voraus und transformiert den Körper, der genug Darstellung für einen Dekoder tragen soll.

UXP behandelt die Datei zuerst als geordnete Inschrift. Es folgt den Unterscheidungen, die die Inschrift lesbar machen, bindet diese Unterscheidungen in eine Normalform und legt eine öffentliche Auslesung dieser Relation frei.

Für das Zehn-Positionen-Profil lautet die intendierte Architektur: Quelleninschrift -> festes relationales Profil -> adressierter Pfad -> rekursive Schalenwechsel -> gebundene spurgegründete Adresslesung -> eine äußere Tiefenverschachtelung -> <10 Ziffern>.uxp -> 0-Byte-Körper -> Gegenlesen nach derselben Regel -> Quelleninschrift

Dies ist nicht „Kompression ohne Nutzlast.“ Es ist ein anderer Darstellungstyp.

Der Anspruch bleibt bedingt durch Exaktheit über der deklarierten zulässigen Quelldomäne. Wenn die Quelle-zu-Öffentlich-Relation auf der erforderlichen Spurklassen-Ebene nicht injektiv ist, ist das Ergebnis kein gültiges, exaktes UXP-Archiv für diese Domäne.

Diese Bedingung gehört in die Theorie hinein, statt einem Kritiker zur Entdeckung überlassen zu werden.

23 Eine Verkleinerung der Darstellung ist keine Kompression

UXP ist eine Verkleinerung der relationalen Darstellung, kein klassischer Kompressionsanspruch. Das Unter-Ziel-Profil verlangt nur:

$width(LNF) \leq 10$.

Es verlangt keine bestimmte zehnstellige Zeichenfolge.

Band IX sagt:

$$G = NF(\Sigma(I))$$

$G \equiv_\Sigma I$.

Die kürzere Relation ist dieselbe Spurklasse an einer anderen Leseadresse. Für das ausführbare Byte-Profil:

$recoverP\ \boldsymbol{L \leq 10(I)} \equiv_\Sigma \Sigma(\gamma I)$, und starke Quellwiederherstellung ist:

$counterP\ \boldsymbol{L \leq 10(I)} = I$. Öffentliche Relationsbreite und die Glyphenlänge einer Zahlennotation sind unterschiedliche Lesungen. Eine hexadezimale oder dezimale Schreibung einer numerischen Adresse kann viele Zeichen enthalten, während sie eine numerische Relationsposition bleibt.

UXP beansprucht daher relationale Readadressierung, keine klassische Reduktion der repräsentationalen Bitanzahl.

24 Der Archiver lehrt den Unterschied, indem er ihn ausführt

Die Einstiegspapiere I und II sind Argumente. Dieser dritte Text ist ein Argument, das ausführbar wird. Das verändert, wie der Tractatus einem Leser vorgestellt werden kann. Ein neuer Leser muss nicht beginnen, indem er das gesamte Vokabular der neun Bände meistert. Der Archiver selbst kann die Inversion lehren.

Zuerst sieht der Leser eine Datei. Dann fragt der Text, was die Datei bereits voraussetzt: Unterscheidungen, Ordnung, Grenzen, Positionen.

Dann wird die Datei als Inschrift umgeschrieben: b0|b1|…|bn.

Dann wird die Inschrift als Pfadrelation gelesen.

Dann wird die Pfadrelation über technische Schnitte hinweg zusammengesetzt. Dann wird dieselbe Relation rückwärts gelesen. Wenn der Leser das Produkt erreicht, ist der fundamentale Zug bereits im Code geschehen. Das Programm ist zu einem pädagogischen Objekt geworden, ohne zu einer Metapher zu werden.

Das ist die Rolle von UXP in der Publikationsreihe.

Einstiegspapier I fragt, warum Mathematik vor axiomatischer Schließung möglich ist. Einstiegspapier II fragt, warum Zerlegung für sich genommen keine vorausgehende Konstruktion beweist. Einstiegspapier III fragt, warum die Behandlung einer Datei als enthaltenes Datenobjekt der Repräsentation bereits vorausgeht. Die verbundenen Inversionen sind: mathematisches Objekt <- Unterscheidung physisches Objekt <- stabilisierte Relation Datenobjekt <- lesbare Inschrift.

Jeder dieser Texte kann unabhängig betreten werden.

Zusammen legen sie dieselbe Bewegung in unterschiedlichen Domänen frei.

25 Der Python- und C-Referenzkern

Python hält die Typentrennung explizit: DigitalNumericalRelation PathTrace RelationalAddress PublicNumericalReadout

C verifiziert dieselbe Ganzpfad-Relation unabhängig.

Version 0.14.1 fügt die Implementierungsunterscheidung hinzu, die direkt aus der Takt-Struktur folgt. Der produktive Dateipfad ist streamingfähig: Quellmarke -> aktuelle Adresse fortsetzen -> nächste Quellmarke -> aktuelle Adresse fortsetzen -> ...

Er konstruiert nicht zuerst eine monolithische Ganzzahl aus der vollständigen Quelle. Der produktive C-Pfad hängt daher nicht mehr von OpenSSL BIGNUM ab.

Der produktive Python-Dateipfad ruft ebenfalls nicht auf: read_bytes() -> int.from_bytes *entire file*

vor der Ausführung.

Für Quellen, die breiter als die Unter-Ziel-Grenze sind, projiziert die aktuelle Demonstration die öffentliche Adresse numerisch in einen hexadezimalen Strom.

Dieser Strom wird inkrementell ausgegeben.

Die hexadezimale Projektion wird nicht zum Adresstyp selbst erhoben.

Die ausführbare Relation bleibt: [I ⟷ _I ⟷Adr_PI ⟷read$Adr_P(I)$.] Es gibt keine Blockkaskade und keine verschachtelte Kindrelation.

Die öffentliche Bedingung lautet nur: UNDER_TARGET = 10 Stop$I$ iff width$I$ <= 10

Die Gegenorientierung empfängt keine ausgeführte Vorwärtsspur, Tiefe, Quelllänge, Pfadhistorie oder Kaskadenzustand.

Die Validierungsberichte zu Version 0.14.1 melden: STREAMING_EXECUTION BLOCK_ONTOLOGY MONOLITHIC_BN_BIN2BN_PRODUCTIVE_PATH UNDER_TARGET

BESTANDEN

16 MiB STREAMING-ROUNDTRIP 64 MiB STREAMING-ROUNDTRIP 128 MiB STREAMING-ROUNDTRIP

BESTANDEN BESTANDEN BESTANDEN

RSS-SPEICHERMAXIMUM 16 MiB RSS-SPEICHERMAXIMUM 64 MiB RSS-SPEICHERMAXIMUM 128 MiB

~4.4 MiB ~4.4 MiB ~4.4 MiB

EXTERNAL_DEPTH EXTERNAL_SOURCE_LENGTH STORED_FORWARD_PATH

Dies klassifiziert das alte 64-MiB-Ergebnis korrekt neu: keine UXP-Grenze keine Readaddress-Grenze kein fehlendes Streaming-Problem sondern: eine Grenze der verworfenen monolithischen Benchmark-Abkürzung

26 Was das aktuelle Produkt behauptet

Zuerst: Die digitale Quelle ist eine geordnete numerische Relation.

Zweitens: Die gesamte Quelleninschrift wird als ein Pfad gelesen.

Drittens: Readadressierung wirkt auf diesen ganzen Pfad, nicht auf Quellteile, die zu kleineren Zahlenentitäten erhoben werden.

Viertens: Spur, Adresse und öffentliche numerische Auslesung sind verschiedene Typen. Fünftens: Es gibt keine feste Zielrelation. UNDER_TARGET = 10 ist nur eine öffentliche Stoppgrenze. Sechstens: Gegenlesen ist dieselbe feste Adressrelation in Gegenorientierung.

Siebtens: Der Gegenpfad schließt bei der festen Referenzrelation, nicht nach einer vorgegebenen Anzahl von Wiederholungen.

Achtens: Keine Quellenlänge, Tiefe, Pfadliste, Blockkarte oder kein zurückbehaltenes Vorwärtsausführungsobjekt ist erforderlich.

Neuntens: Jede endliche Byte-Inschrift im Referenzbereich hat eine erzeugte Ganzpfadadresse und einen exakten Gegenpfad zur festen Wurzel.

Zehntens:

Öffentliche Relationsbreite ist nicht die Glyphenzahl einer Zahlennotation, die zur Anzeige einer numerischen Adresse verwendet wird.

Für:

$$I = b0 \mid \cdots \mid bn-1$$

sei:

$a0 = \odot \varphi$,

$aj+1 = N\ extP\ (aj\ ,\ bj\ )$.

Dann:

$L(I) = read(an\ )$ und:

$counterP\ (L(I)) = I$.

Für $n > 10$ legt die aktuelle Referenz die vollständige Adresse als eine öffentliche numerische Position offen:

$$width(L(I)) = 1 \leq 10$$

27 Was das aktuelle Produkt nicht bedeutet

Die globale Pfadresse ist kein Blockkompressor.

Es bedeutet nicht: Quelle -> Gruppen -> Adressen von Gruppen -> verschachtelter Baum

Es bedeutet nicht, dass öffentliche Zahlennotation die Spur selbst ist.

Es bedeutet nicht, dass eine externe Quellenlänge dem Programm sagt, wann es anhalten soll. Es bedeutet nicht, dass ein früherer Lauf per Adresse nachgeschlagen wird. Und es bedeutet nicht, dass die öffentliche Auslesung höchstens zehn Textzeichen hat. Die Unter-Ziel-Bedingung lautet:

$width(L) \leq 10$, nicht:

$glyphcount$(one numeral spelling) ≤ 10.

Die Typunterscheidung bleibt bestehen:

$\Sigma \neq Adr \neq read(Adr)$.

Eine weitere Implementierungsdrift muss ebenfalls ausgeschlossen werden: Readaddress-Relation -> öffentliche Zahlenauslesung -> die Zahl als riesige lineare Hex-Zeichenkette serialisieren -> die Größe dieser Zeichenkette als Größe der Relation behandeln

Der letzte Schritt ist ungültig.

Die hexadezimale Zeichenkette ist eine Schnittstellen-Projektion von readAdr. Ihre physische Größe kann für die Schnittstellen-Entwicklung relevant sein, aber sie bestimmt rückwirkend weder die relationale Breite noch die Streaming-Struktur des Vollzugs. Also: [Streaming-Vollzug ≠/lineare Serialisierung einer Zahlennotation.]

28 Der minimale Beweisrücken

- 1. Digitale numerische Inschrift

$$I = b0 \,|b1 \,| \cdots |bn{-}1$$

- 2. Ganzer Pfad

$I \mapsto \gamma I$.

- 3. Festes Profil und Wurzel

$P = (Adr, N\,ext, \varphi, N\,F\,, D, U\,)$,

$U = 10$,

$a0 = \odot\ \varphi$.

- 4. Pfadfortsetzung

$aj{+}1 = N\,extP\,(aj\,, bj\,)$.

- 5. Ganzpfadresse

$AdrP\,(I) = an$.

- 6. Öffentliche Auslesung

$L(I) = read(AdrP\,(I))$.

- 7. Unter-Ziel

$StopU\,(L) \Longleftrightarrow width(L) \leq 10$.

- 8. Dieselbe Relation in Gegenorientierung

$P\,redP\,(aj{+}1\,) = (aj\,, bj\,)$.

Wiederholtes Gegenlesen erreicht die feste Wurzel ohne Quellenlängenparameter.

- 9. Exaktheit

$counterP\,(L(I)) = I$.

Auf Spurebene:

$recoverP\,(L(I)) \equiv\Sigma\ \Sigma(\gamma I\,)$.

Der ausführbare Rücken ist:

$$| \rightarrow I \rightarrow \gamma I \rightarrow AdrP \rightarrow read$$

und in Gegenorientierung:

$read \rightarrow AdrP \rightarrow \gamma I \rightarrow I$.

29 Von Difference zu ausführbarer Difference

Der Tractatus Principio wird nicht dadurch zu Software, dass philosophische Sätze in Klassennamen übersetzt werden.

Band VIII gibt die operative Sequenz:

$I \rightarrow \gamma \rightarrow \Sigma \rightarrow \Pi \rightarrow \bowtie \rightarrow Lock \rightarrow N\,F \rightarrow read$.

Die Implementierungsaufgabe besteht darin, diese Typen zu bewahren und dabei gewöhnliche Maschinenrepräsentationen zu verwenden. Die tiefere Bewegung ist:

$$| \leftrightarrow A|B \leftrightarrow \Delta = |(|) \leftrightarrow Adr \leftrightarrow \gamma \leftrightarrow \Sigma \leftrightarrow \Pi \leftrightarrow N\ F \leftrightarrow read$$

Der Code führt diese Lesungen nacheinander aus, aber die drei Ordnungen von Difference sind keine zeitlichen Stufen.

Daraus folgen drei Implementierungsdisziplinen.

Erstens: Projektion ist kein Trägerzustand.

Eine sichtbare Ziffer ist nicht die Adresse.

Zweitens: Tiefe ist eine Leserolle, keine pro-Datei-Kernel-Fracht.

Band IX stellt fest:

$depth \notin KernelState$, Drittens:

$depth := readd\ \boldsymbol{\Sigma}$.

Schließung ist relational, kein externes Gegenlese-Argument.

Viertens: Eine reversible Integer-Zustandsmaschine ist noch keine UXP-Spur.

Eine Funktion kann bijektiv sein und dennoch den falschen Typ haben, wenn sie:

$I \rightarrow \gamma \rightarrow \Sigma \rightarrow \Pi \rightarrow \bowtie \rightarrow Lock \rightarrow N\,F \rightarrow read$ ersetzt durch:

$integer \leftrightarrow integer$.

Die finale Laufzeit darf sich daher nicht auf einen versteckten Dateigrößenparameter stützen, um das Gegenlesen zu stoppen. Ebenso wird Δ nicht als zusätzlicher Kompressionsoperator eingeführt:

$$\Delta := |(|)$$

Hüllenwechsel, Bindung, Rotation, Normalisierung und Gegenlesen sind operative Lesungen relationaler Selbstreferenz.

Die Kodierdisziplin lautet: Textrelation -> typisierte operative Relation -> ausführbare feste Regel -> Test im frischen Prozess

nicht: vertraute Speicherstruktur -> versteckte Reparatur -> Reinterpretation als UXP

30 Die Wirkung des Archiver

Der Archiver macht die grundlegende Behauptung operativ falsifizierbar.

Die beabsichtigte Anwendungsgrenze ist: eine Quelle innerhalb des deklarierten zulässigen Bereichs wählen sie als geordnete Inschrift lesen ihren Pfad bilden ihre Spur integrieren Tiefe/Adresse/Orientierung aus dieser Spur lesen die kompatiblen Projektionen binden Lock_3/4 erkennen die Normalform bilden die vollständige öffentliche Auslesung nehmen <öffentliche Auslesung>.uxp schreiben

den Körper leer lassen einen frischen Prozess mit demselben festen Profil starten die öffentliche Normalform unter P gegenlesen derselben Relation in Gegenorientierung folgen Lock_P erreichen die Quelleninschrift wiederherstellen

Die Architektur legt daher ihre eigenen Testpunkte offen.

Wo ist der Quellblock?

Nicht im .uxp-Körper.

Wo ist der gespeicherte Pfad?

Es gibt kein pro-Datei-Pfad-Seitenobjekt.

Wo ist Tiefe?

Sie ist:

readd Σ.

Sie wird nicht aus einem dezimalen Metadatenfeld entpackt.

Wo ist Adresse?

Sie ist:

readAdr Σ.

Sie ist nicht der Kernel-Träger und nicht Adresse % 10.

Wo ist Quellenlänge?

Sie darf nicht extern geliefert werden. Schließung muss aus der Relation gelesen werden. Was ist dateispezifisch?

Die öffentliche Normalform-Auslesung.

Was ist gemeinsam?

Das feste ausführbare Profil, einschließlich seiner Eichung.

Was begrenzt Exaktheit?

Der deklarierte zulässige Bereich, auf dem die öffentliche Auslesung eindeutig gegengelesen werden kann. Dies sind keine Implementierungsfußnoten. Sie sind der operative Inhalt der Theorie.

31 Warum das Produkt wichtiger ist als das Entwicklungstagebuch

Eine Veröffentlichung sollte die invariante Architektur darstellen und den Leser nicht zwingen, sie aus einem Entwicklungstagebuch zu rekonstruieren.

Die relevante Frage ist daher nicht, welcher Prototyp zuerst erschien.

Sie lautet, welche Implementierungstypen mit dem UXP-Vertrag kompatibel sind. Die folgenden Punkte sind unter diesem Vertrag Kategorienfehler:

• die öffentliche Zahl als Container für verstecktes Quellmaterial behandeln; • die Zielkaskade zu einer skalaren Rekurrenz verflachen; • die Pfad-/Spurordnung durch eine reversible Integer-Zustandsmaschine ersetzen; • den Kernel-Träger mit einer endlichen dezimalen Adressoberfläche identifizieren; • gleiche sichtbare Label mit identischen relationalen Positionen identifizieren; • die sichtbare Marke als Rest einer gepackten Adresse berechnen; • eine beliebige affine Permutation so behandeln, als ob ihre Reversibilität UXP-Gegenlesen begründete; • Arbeiterschnitte mit Archivtiefen verwechseln; • eine partielle Projektion mit dem vollständigen Normalform-Label identifizieren; • Tiefe als persistierten pro-Datei-Kernel-Zustand behandeln; • Adresse, Rotation oder Tiefe als unabhängige Nutzlastfelder behandeln; • die Quellenlänge während des Gegenlesens extern liefern; • einen Vorwärts-Knotengraphen speichern und das Nachschlagen „Counterreading“ nennen; • verschiedene Adressen zwingen, auf einer Hülle zu kollidieren, statt die Referenzebene zu wechseln; • das Ausführungsprofil ändern, nachdem das gewünschte Ergebnis beobachtet wurde; • rohe verschachtelte Kombinationen zählen und dabei den profil-zulässigen Bereich ignorieren; • beliebige Quellenabdeckung behaupten, bevor dieselbe Relation sie in beiden Orientierungen etabliert hat.

Die positive Architektur ist: Quelleninschrift -> Pfad -> Spur -> Nur-Lese-Projektionen -> Bindung -> Lock_3/4 -> Normalform -> öffentliche Auslesung -> Artefakt mit leerem Körper

und in Gegenorientierung: öffentliche Auslesung -> Normalform-Spurklasse -> dieselbe Relation in Gegenorientierung -> Quelleninschrift

Tiefe und Adresse werden aus der Spur gelesen, an welchem Punkt auch immer sie erforderlich sind. Diese Darstellung macht die stärksten Einwände zu einem Teil der Spezifikation selbst. Ein Kritiker muss dem Vokabular nicht vertrauen.

Der Kritiker kann testen, ob die Implementierung den Typvertrag einhält.

32 Das Argument fortsetzen

Dieser Text ist der dritte Einstiegspunkt in den Tractatus Principio. Das erste Einstiegspapier fragt, was Mathematik bereits voraussetzt, bevor ein Axiom funktionieren kann: C.H.F. Warnke, All You Need Is Difference — Why Mathematics Works — and Still Fails Reality. Das zweite fragt, ob Zerlegung eine vorgängige ontologische Konstruktion begründet: C.H.F. Warnke, The Illusion of Construction — Why There Is No Matter.

Der vorliegende Text trägt dieselbe Umkehrung in ausführbare Information. Zusammen können die drei Einstiegspunkte geschrieben werden: mathematisches Objekt <- Unterscheidung physisches Objekt <- stabilisierte Relation Datenobjekt <- lesbare Inschrift.

Sie sind unabhängige Eingänge, keine vorgeschriebene Leseordnung.

Ein Leser kann hier mit dem Archiver beginnen und rückwärts zur Differenz gehen. Ein anderer kann mit der Differenz beginnen und hier als Anwendung ankommen. Die rekursive Verbindung ist der Punkt.

33 Schluss

Kompression funktioniert.

Darum darf der Titel nicht missverstanden werden.

Kompression ist nur dann eine Sackgasse, wenn die Grundfrage auf die endlose Suche nach einem kleineren Container für ein bereits vorausgesetztes Informationsobjekt reduziert wird. UXP verändert den Gegenstand der Erklärung.

Eine Datei ist zuerst eine lesbare Inschrift. Ihre Ordnung begründet eine Pfadrelation. Aufeinanderfolgende Relationen komponieren, ohne technische Schnitte in mathematische Schnitte zu verwandeln. Perspektive beschränkt, welche Fortsetzungen zulässig sind. Kompatible relationale Zwänge verengen die Kandidatenmenge. Der dezimale Träger zeigt bereits dieselbe Struktur, bevor die größere Kaskade beginnt. Sein operativer Ursprung ist keine Tabelle von zehn verschiedenen Zahlobjekten. Er ist eine Selbstreferenz:

$$\Delta = |(|)$$

gelesen in einem geordneten Takt.

Für einen endlichen Lauf:

$$\Delta[n] = T1(\Delta)|\cdots|Tn(\Delta)$$

bleibt die Relation selbstisomorph zum selben Delta-Ursprung:

$$\Delta[n] \equiv_\Delta \Delta$$

Die Zahl ist die öffentliche Umschreibung dieses Takts.

Das Dezimalsystem synchronisiert diese Lesungen auf einer Hülle:

```
0|1|2| ⋯ |9
```

Die 9 ist die sichtbare Bruchkante der Hülle.

Tiefe beginnt erst, wenn die bereits differenzierte Taktordnung selbst zur Referenz einer weiteren Unterscheidung wird.

Die Dezimalhülle zeigt weiterhin, dass Zahl eine Auslesung einer selbstrelationalen Taktordnung ist. Der Archiver verlangt jedoch nicht mehr, dass eine Quelle als abgeschlossene dezimale Hüllenblöcke ankommt. Die ganze digitale numerische Relation:

$$I = b0 \ |b1 \ | \ \cdots \ |bn-1$$

wird als ein Pfad gelesen:

$I \rightarrow \gamma I$.

Unter dem festen Profil:

$a0 = \odot \ \varphi$,

$aj+1 = N \ extP \ (aj \ , \ bj \ )$.

Die erreichte Adresse ist:

$AdrP \ (I) = an$, und die öffentliche Auslesung ist:

$L(I) = read(AdrP \ (I))$.

Es gibt keine feste Zielzahl.

Es gibt nur:

$StopU \ (L) \Longleftrightarrow width(L) \leq 10$.

Dieselbe Relation counter-reads die Adresse zur festen Wurzel zurück und reproduziert die ursprünglichen geordneten Marken:

$counterP \ (L(I)) = I$.

Keine Quellenlänge, Tiefe, kein Blockbaum und keine frühere Ausführung wird geliefert. Dies ist der operative Inhalt von: Die Regel ist der Pfad.

Der Pfad wird nicht als weiteres Objekt gespeichert, weil der zulässige Pfad durch die Relation erzeugt wird.

Die sichtbare Zahl ist nicht primär. Sie ist eine öffentliche Auslesung einer relationalen Form. Ein Zahlensystem ist die gemeinsame Leseebene, auf der solche Formen gleichzeitig lesbar werden.

Tiefe ist keine Nutzlast, keine Zählung von Bruchkanten-Glyphen und nicht die Klammerntiefe einer gewählten Zahlzerlegung. Sie ist eine Lesung wirksamer Selbstreferenz in der Spur. Komplexität ist kein Haufen zusätzlicher Dinge. Sie ist zunehmende relationale Bestimmung.

Einstiegspapier I fragt, was Mathematik bereits benötigt, bevor ein Axiom funktionieren kann.

Einstiegspapier II fragt, ob Zerlegung vorgängige Konstruktion beweist.

Einstiegspapier III fragt, ob Information zuerst als enthaltenes Datenobjekt behandelt werden muss, bevor sie dargestellt werden kann.

Die drei Umkehrungen bleiben: mathematisches Objekt <- Unterscheidung physisches Objekt <- stabilisierte Relation Datenobjekt <- lesbare Inschrift

Kompression verändert die Inhalte der Box.

UXP verändert die Lesung - und lässt die Box leer.

Referenzen

Warnke, C.H.F. All You Need Is Difference - Why Mathematics Works - and Still Fails Reality. The Future Foundation, Entry Paper No. 01, 2026.

Warnke, C.H.F. The Illusion of Construction - Why There Is No Matter. The Future Foundation, Entry Paper No. 02, 2026.

Warnke, C.H.F. Tractatus Principio - Über den Unterschied. Band I: Buch der Mathematik. Insbesondere der Primat von Difference, die relationale Lesung von Zahl und die Untrennbarkeit von Zahl und geometrischer Spur.

Warnke, C.H.F. Tractatus Principio - Über den Unterschied. Band IV: Buch der Selbstbezüglichkeit. Insbesondere die gleichzeitigen Ordnungen $|, A|B$ und $|(|)$ und die Definition $\Delta := |(|)$. Warnke, C.H.F. Tractatus Principio - Über den Unterschied. Band VI: Buch der Adressierung. Insbesondere §§1-6 zu Hülle, Tiefe, Adresse und Fixierung sowie §§15-16 zur δ-Verfeinerung und Schließung unter Fixierung.

Warnke, C.H.F. Tractatus Principio - Über den Unterschied. Band VIII: Buch der Anwendung. Insbesondere die ausführbare Relation $I \to \gamma \to \Sigma \to \Pi \to \bowtie \to Lock \to NF \to read$ und die Anforderung, dass Code Pfad, Bindung, Spur, Adresse und Wiederherstellbarkeit bewahrt. Warnke, C.H.F. Tractatus Principio - Über den Unterschied. Band IX: Buch der Lesung. Insbesondere §4.3 zu gültigen Labels, §§5-6 zu Hüllenmarken und Inschrift, §§9-10 zu Tiefe und Projektion, und §§11-15 zu Bindung, Lock, Normalform und Readaddressing.

Warnke, C.H.F. Das kartesische Koordinatensystem als Delta-Lesearchitektur. Insbesondere die Typtrennung zwischen Adresse, Spur, Projektion und öffentlicher Auslesung; $C\varphi\ \boldsymbol{A}, \boldsymbol{\Sigma} = (d, \ell, q)$; und die starke Wiederherstellungsbedingung für vollständige Normalform-Labels.

Hinweis zum Geltungsbereich Diese Version 3.7 präsentiert die Korrektur Streaming-Vollzug / Schnittstellen-Projektion der globalen Readaddress-/Unter-Ziel-Architektur.

Der ausführbare Referenzbereich bleibt:

$$D_P =_{n \geq 0} 0, \ldots, 255^n.$$

Die zentrale Korrektur ist kein neuer mathematischer Mechanismus.

Sie ist die Entfernung einer unnötigen Implementierungsnormalisierung: BN_bin2bn$entire_file$

aus dem produktiven C-Pfad.

Die Relation bestimmt bereits die inkrementelle Ausführung: [a_{j+1}=Next_Pa_j, b_j.] Daher: streaming_execution = BESTANDEN streaming_is_new_architecture = NEIN block_ontology = 0 monolithic_whole_file_bignum = 0

Korrigierte C-Validierung in frischen Prozessen: 16 MiB exakter Roundtrip = BESTANDEN 64 MiB exakter Roundtrip = BESTANDEN 128 MiB exakter Roundtrip = BESTANDEN RSS-Speichermaximum über 16/64/128 MiB ≈ 4.4 MiB

Der frühere 64-MiB-BN_bin2bn-Fehlschlag wird als UXP-Skalierungsaussage zurückgezogen. Die verbleibende Implementierungsfrage betrifft die aktuelle öffentliche Schnittstellen-Projektion: Eine adressierte numerische Position wird weiterhin als vollständige lineare hexadezimale Zahl geschrieben. Diese Notation wird absichtlich klassifiziert als: Schnittstellen-Projektion

und nicht als: Relationsbreite Adressontologie Streaming-Architektur

Künftige Schnittstellenarbeit muss Wiederherstellbarkeit bewahren, ohne versteckten dateispezifischen Zustand, Blockontologie oder eine Kodierer/Dekoder-Spaltung wieder einzuführen.

Addendum: Das Byte als getaktete Relation von Difference

Eine werkimmanente Erweiterung zur dateibezogenen Medium-Lesung von Publication III.

Status dieses Addendums. Das vorausgehende Argument bleibt unverändert. Dieses Addendum erweitert die Medium- und Dateidiskussion, indem es ausdrücklich macht, wie ein physisches Byte gelesen werden kann, ohne Bits, Bytes oder Maschinen-Integer zur Ontologie von UXP zu machen.

Die Erweiterung bleibt unter derselben Autoritätsrichtung, die der Text bereits festgelegt hat: Difference vor Zahl, Relation vor Wert, die Regel als Weg und die öffentliche Auslesung als Leseadresse statt als Container.

A.1 Das Byte vor dem Wert

Ein Byte soll nicht zuerst als fertiger Skalarwert `0` bis `255` in die Tractatus-Relation eintreten. Dieser Skalar ist bereits eine stabilisierte Maschinenlesung eines achtstelligen physischen Trägers.

An der Maschinengrenze kann das Byte stattdessen als acht binärseitige Positionen gelesen werden:

```
B = b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0
bj in {0,1}
```

Die Notation `{0,1}` darf nicht zu zwei primitiven Substanzen verdinglicht werden. Die beiden Marken sind lesbare Positionen einer Unterscheidung. Die `1` ist die markierte Position. Die `0` ist nicht die Abwesenheit von Relation. Sie ist die geschlossene oder unmarkierte Gegenposition im

selben relationalen Feld.

```
0 != nichts
```

Darum ist `00000000` kein leerer Nicht-Zustand. Es ist eine vollständige achtstellige Konfiguration, in der jede aktuelle Position unter der geschlossenen Seite der Unterscheidung gelesen wird.

A.2 256 Konfigurationen, 255 als höchste Auslesung

Acht binärseitige Positionen haben einen endlichen Konfigurationsraum. Die bekannte Zählung ist hier nützlich, sofern sie nicht in eine Speicherontologie verwandelt wird:

```
2^8 = 256 Konfigurationen
```

Die konventionelle vorzeichenlose numerische Oberfläche liest diese 256 Konfigurationen dann als Folge:

```
0, 1, 2, ..., 255
```

Der Unterschied zwischen `256` und `255` ist daher typologisch, kein fehlender Zustand. `256` benennt die Anzahl verfügbarer achtstelliger Konfigurationen. `255` ist der höchste Wert der konventionellen nullbasierten Skalarlesung desselben endlichen Trägers.

Die All-Null-Konfiguration zählt. Ihre numerische Auslesung ist `0`. Auch die All-Eins-Konfiguration zählt. Ihre konventionelle vorzeichenlose Auslesung ist `255`. Keine der beiden Konfigurationen ist ontologisch vollständiger als die andere.

```
256 = Konfigurationsanzahl 255 = höchste konventionelle vorzeichenlose Auslesung
```

Diese Unterscheidung ist für UXP wichtig, weil die kombinatorische Oberfläche nicht mit kleinen Informationsobjekten verwechselt werden darf, die im Byte gespeichert seien. Das Byte ist ein Maschinenträger, auf dem eine endliche relationale Konfiguration öffentlich lesbar wird.

A.3 Das Byte als Selbstrelation der Unterscheidung

Die acht Positionen sind nicht acht isolierte Bit-Dinge, die später zu einem Byte-Objekt zusammengesetzt werden. Jede neue Position tritt in eine Relation ein, in der die vorhergehenden Unterscheidungen bereits wirksam sind.

```
Sigma_j | nächste Position -> Sigma_(j+1)
```

So ist `1|1` nicht bloß zweimal dasselbe Objekt. Die zweite markierte Position ist relativ zu einer bereits wirksamen markierten Position unterschieden. Der nächste Takt wirkt daher auf eine Relation.

```
1 -> 1|1 -> (1|1)|1 -> ...
```

Der achtstellige Träger kann entsprechend als endliche Stabilisierung wiederholter Selbstreferenz der Unterscheidung gelesen werden. Die Selbstreferenz wird nicht hinzugefügt, nachdem die Bits gesammelt wurden. Sie macht deren geordnete Relation überhaupt erst wirksam.

```
Delta := |(|)
```

In diesem Sinn ist das Byte eine kleine, endliche, maschinenseitige Lesung derselben Bewegung, die bereits den Tractatus bestimmt: Difference unterscheidet ihre eigenen bereits wirksamen Unterscheidungen.

A.4 Strich, Punkt und orthogonale Lesung

Die binärseitige Position darf ebenfalls nicht auf das naive Paar „Strich“ gegen „kein Strich“ reduziert werden. Die Nullposition ist nicht nichts, und der Strich selbst lässt mehr als eine öffentliche Orientierung zu.

Ein quer gelesener Strich kann frontal als Punkt gelesen werden, wenn die Perspektive orthogonal gedreht wird. Der Punkt ist kein neu erzeugtes Objekt, und der Strich ist nicht verschwunden. Dieselbe Relation wird von einer anderen Face gelesen.

```
| <-> Punkt unter i = R_90
```

Die imaginäre Achse liefert daher die operative Lesung, durch die sichtbare Ausdehnung zu Tiefe werden kann, ohne die Trägerrelation zu verändern. Die orthogonale Lesung ist keine zweite Substanz und kein zusätzlicher Informationskanal.

Für das Byte verhindert dies eine weitere verbreitete Verdinglichung. Eine physische Bitposition ist kein ontologischer Miniaturschalter, dessen Wahrheit in einem Skalar enthalten wäre. Sie ist eine maschinensichtbare Position innerhalb einer Relation, die auch orthogonale Lesung zulässt.

A.5 Von acht physischen Bitpositionen zu zwei hexadezimalen öffentlichen Marken

Das aktuelle ausführbare `P_UXP`-Profil muss Binär nicht wieder als aktive öffentliche UXP-Hülle einsetzen. Physische Bitspeicherung kann Eigenschaft des Maschinenträgers bleiben, während die dateibezogene öffentliche Hülle hexadezimal oder dezimal ist.

Für byteorientierte Computerdaten liefert Hexadezimal eine direkte öffentliche Leseoberfläche: Ein physisches Byte ist öffentlich als zwei hexadezimale Marken lesbar.

```
8 physische Bitpositionen <-> 2 hexadezimale öffentliche Positionen
```

Zum Beispiel kann das Maschinenmuster `10110010` auf der hexadezimalen öffentlichen Oberfläche gelesen werden als:

```
B|2
```

Dies darf nicht in die Behauptung übersetzt werden, `B2` enthalte `10110010`, oder `10110010` sei das verborgen wahrere Objekt hinter `B2`. Es sind verschiedene öffentliche Lesungen desselben physischen Trägers unter verschiedenen Lesekonventionen.

```
10110010 ~= B|2 als öffentliche Lesungen eines Trägers
```

Hexadezimal genügt daher als dateibezogene Maschinenhülle für Byte-Muster, ohne das Byte zuerst in ein dezimales Skalarobjekt zu verwandeln.

A.6 Dateibezogene Konsequenz

An der Betriebssystemgrenze kann ein Programm ein Byte in einem gewöhnlichen Maschinentyp wie `unsigned char` empfangen. Diese Implementierungstatsache muss der Relation nachgeordnet bleiben.

Der vom Maschinentyp zurückgegebene Wert — zum Beispiel `178` für den Träger, dessen hexadezimale Oberfläche `B2` ist — darf nicht stillschweigend zur mathematischen Bedeutung des Bytes innerhalb von UXP werden.

Die dateibezogene Lesung ist daher so zu verstehen:

```
physischer Byte-Träger -> zwei aktuelle hexadezimale öffentliche Positionen -> Takt -
> aktuelle Relation -> P_UXP -> read(Adr)
```

Die zwei hexadezimalen Positionen sind kein Zwei-Slot-Paket, das acht versteckte Bits enthält. Sie sind die gewählte öffentliche Hüllenlesung desselben aktuellen Trägers.

Ebenso wird ein begrenzter I/O-Puffer nicht zu einer Byte-Block-Relation. Der Puffer legt nur die nächsten physischen Träger für denselben geordneten Vollzug frei. Die lebendige Relation bleibt eine.

Diese Erweiterung stärkt daher die bereits in Publication III formulierte Streaming-Regel: Der nächste maschinensichtbare Träger wird in die gegenwärtige Relation gelesen und muss nicht als früheres Byte-Objekt, Bit-Objekt oder Skalarhistorie archiviert werden.

A.7 Profil-Spezialisierung

Dieses Addendum unterscheidet einen physischen Maschinenträger vom aktiven öffentlichen UXP-Hüllenprofil.

Physische Maschinen-Tatsache: heutige byteadressierte Maschinen tragen physisch Bytes, deren Zustand durch acht binärseitige Positionen beschrieben werden kann.

Aktuelles ausführbares UXP-Profil: Hexadezimal und Dezimal sind die aktiven öffentlichen Hüllen. Binär wird hier nicht als zusätzliche öffentliche UXP-Hülle wiedereingeführt.

Die Rolle der physischen Bitbeschreibung ist daher erklärend und dateibezogen: Sie zeigt, warum Maschinenbytes in die hexadezimale öffentliche Hülle gebracht werden können, ohne ein Byte als skalaren Informationscontainer zu behandeln.

A.8 Implementierungsleitlinie für Bytes

Die Byte-Erweiterung ist nur dann treu, wenn die folgenden Unterscheidungen ausdrücklich bleiben:

```
physischer Bit-Träger != binäre Ontologie Byte != Skalarobjekt 0..255 00000000 !=
Abwesenheit von Relation 256 Konfigurationen != höchste Auslesung 255 zwei
hexadezimale Marken != Container für acht versteckte Bits unsigned char !=
mathematisches UXP-Primitiv hexadezimale öffentliche Lesung != Kompressionscode
Puffergrenze != relationale Grenze
```

Eine treue Implementierung darf Maschinenbytes, Nibble-Extraktion, begrenzte Puffer, Integer-Register und gewöhnliche Datei-APIs verwenden. Diese Träger sind erlaubt, weil Implementierungskomfort der relationalen Lesung nachgeordnet bleibt.

Verboten ist die Umkehrung der Autorität, in der der Maschinenskalar `0..255` zur Ontologie wird, aus der die UXP-Relation anschließend rekonstruiert wird.

A.9 Schlussformel

Das Byte enthält keine acht primitiven Informationsdinge. Es ist ein endlicher physischer Träger, auf dem eine Unterscheidung über acht geordnete Positionen lesbar wird.

Die Nullposition ist nicht nichts. Die Einsposition ist kein isoliertes Ding. Die geordnete Konfiguration ist eine getaktete Selbstrelation von Difference.

```
Bitposition = binärseitige Lesung einer Unterscheidung Byte = achtstellige
Stabilisierung dieser getakteten Relation Hex = öffentliche Hüllenlesung desselben
Byte-Trägers
```

Die Maschine darf den Träger physisch speichern. UXP macht Speicherung dadurch nicht zur Ontologie der Relation.

Das Byte ist der Ort, an dem eine endliche Maschinenrelation lesbar wird. Die hexadezimalen Marken sind der Ort, an dem dieser Träger unter der aktuellen dateibezogenen Hülle öffentlich lesbar wird.

Die Regel bleibt fest. Der Weg wird vollzogen. Die Relation wird gelesen.